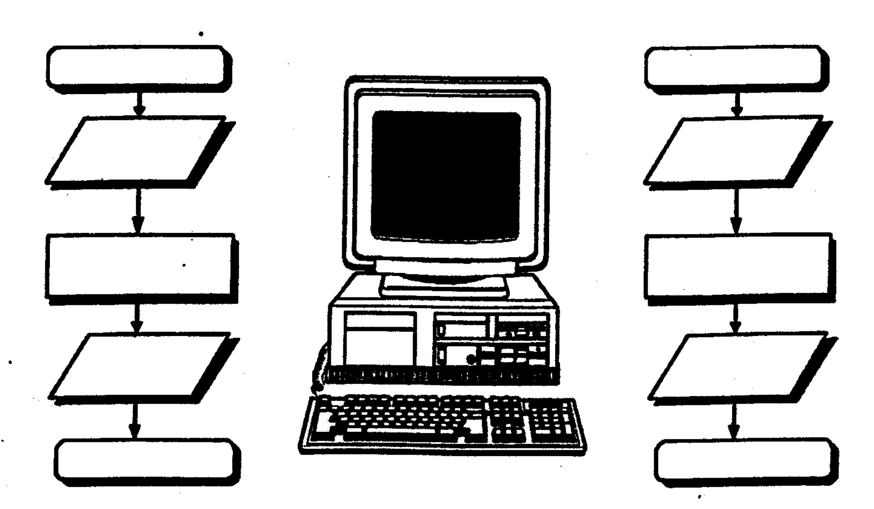
سلسلة الحاسب الآلي وتكنولوجيا المعلومات

(٨)

البرمجة باستخدام Visual Basic 6

Microsoft



الدكتور

محمد محمد رفعت البسيوني

أستاذ إستخدامات الحاسب الآلي في التعليم المساعد

قسم إعداد معلم الحاسب الآلي

كلية التربية النوعية بدمياط

جامعة المنصورة

رقم الإيداع بدار الكتب والوثائق القومية ٥٥١٤ / ٢٠٠٦



بسم الله الرحمن الرحيم

" وعلمك ما لم تكن تعلم وكان فضل الله عليك عظيماً "

صدق الله العظيم

سورة النساء (آية ١١٣)

تمهيد:

كان الدافع لإعداد هذا الكتاب الحاجة إلى متابعة التطورات الحديثة التي تدخل يوميا على مجال الحاسبات الآلية. وبالتحديد متابعة التطورات الخاصة بالبرمجيات الجاهزة، وخصوصاً البرامج الجاهزة التطبيقية، مثل برنامج البيسيك المرئي Visual Basic، والتي يمكن أن تستخدم لحل كثير من المشاكل والتي تحتاج لمرونة ما في التعامل وشكل أيسر في الاستخدام لغير المتخصصين، لكونها برمجة ذات سمات جديدة، وتمتاز بكثير من المميزات سوف تلمس عند التعرف علي هذا النظام.

ويحتوي هذا الكتاب علي الأساسيات الضرورية لتعلم البيسيك المرئي VISUAL BASIC، وسيتم التركيز علي ذلك من خلال فصول هذا الكتاب، ويحتوى هذا الجزء على المبادئ الأساسية الخاصة بتعلم كيفية استخدام برنامج VB وإعطاء بعض النماذج او الأمثلة البسيطة للاستخدام وإعداد البرامج لحل مشاكل مختلفة.

وهذا البرنامج تم إعداد بواسطة شركة Microsoft الأمريكية للبرمجيات، وذلك للعمل على الأجهزة الشخصية لشركة IBM - PC ، وبالتحديد أيضا للعمل من خلال بيئة برامج التشغيل النوافذ WINDOWS.

وقد ظهر أول إصدار لهذا البرامج في عام ١٩٨٧ وكان للإصدار الأول Ver. 2.0 ثم تلاه الإصدارات الأخرى لهذا البرنامج رقم ٣ و ٤ و ٥ والإصداررقم Ver 6.0 في عام ١٩٩٨ والذي سيدور حوله هذا الكتاب، ثم تلي ذلك ظهور VB DOT NET والخاص ببرمجة الويب.

وسوف تعرض المادة العلمية الخاصة بهذا الكتاب في صورة مجموعة من الفصول يجب على القارى لها ان يتابع التسلسل المنطقي في الاستخدام لها وفهم محتوياتها حتى يستطيع ان يتعرف بشكل جيد على كيفية استخدام برنامج VB6.

وهذه المادة العلمية سوف يستفاد منها ذوى الخبرة في البرمجة من حيث توظيف إمكانيات البيسيك المرئي في خدمتهم، حيث ستوفر لهم كثير من الوقت والمجهود في اعداد برامجهم.

والله ولي التوفيق.

مؤلف الكتاب

الفصل الأول

مدخل إلى البيسيك المرئي Visual Basic

يتناول هذا الفصل المحتويات التالية:



١ - ١ تمهيد :

بدأت Microsoft فى تطوير هذا البرنامج ليلائم التطورات المصاحبة لبرنامج نظام التشغيل النوافذ WINDOWS ، وظهر الإصدار الثاني لهذه اللغة Version 2.0 كبرنامج مستقل ذو كيان خاص به من حيث الإعداد داخل بيئة النوافذ وأيضا التشغيل ثم الإمكانيات الكبيرة لمعالجة المشاكل البرمجية، ثم تلا ذلك إصدارات جديدة وصولا إلي الإصدار Version 6.0 كإصدارات متطورة لوضع هذا البرنامج موضع اللغات القوية القريبة الشبه من لغات الآلة أو لتصبح لغة قوية لتحاكى الآلة مباشرة، وبالتالي ستكون البرامج المعدة من خلالها ذات مستوى عالي.

يعتبر هذه البرنامج الطريق الأسرع والأسهل في تصميم البرامج أو فى تخطيط البرامج. حيث يتم من خلالها إنشاء البرامج بتكلفة أقل من نظيراتها من حيث الوقت والحيز المستخدم وأيضا تكوين الأوامر داخل البرنامج بسهولة لتؤدى وظائف كبيرة وقوية لتوفر فى عدد الأوامر المستخدمة من قبل في أي لغة أخر، بالإضافة إلى أنه أمكن البرمجة باستخدام اللغة العربية كلغة وسيطة لتسجيل البيانات مثلها مثل البرمجة باستخدام برامج ACCESS و EXCEL، وبالتالي تم حل كثير من المشاكل المرتبطة بالمستخدم العربي الذي لا يجيد التعامل مع الحاسبات الآلية وبالأخص برمجة الحاسب الآلي.

ويمتاز أيضا هذا البرنامج بسهولة اكتشاف الأخطاء وتصحيحها، بالإضافة إلى إمكانية الترجمة التنفيذية للبرامج المصدرية ،أي أن الملف يكون قابل للتنفيذ Executable في ثوان معدودة، ويمكن القول أن تخطيط البرامج بهذا البرنامج في الحقيقة يشابه تخطيط البرامج باستخدام لغة C.

وبالتالي يمكن الاستفادة بالمميزات التي تمنحها البرمجة هذه مثل تشغيل برنامج كامل من خلال الضغط على رمز أو زر موجود على واجهة البرنامج الخاص بك، وبالتالي ما عليك إلا تحريك المؤشر إلى هذه النافذة أو الرمز والنقر عليه لتشغيل برنامج معين بدون مجهود يذكر.

١ - ٢ كيفية تهيئة (إعداد) برنامج Microsoft Visual Basic 6.0 :

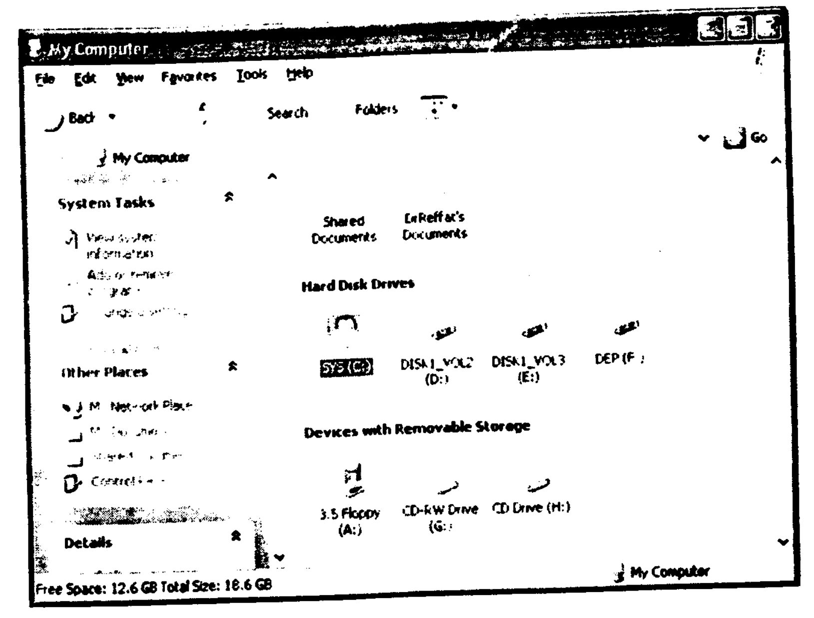
يوجد البرنامج على اسطوانة ليزر ضمن المجموعة المسماة Microsoft Visual Studio 6.0 باسم Microsoft Visual Basic 6.0، وبالتالي يجب على المستخدم التعامل مع وحدة إدارة الأسطوانات المضغوطة بجهاز الحاسب الآلي لتنفيذ عملية التهيئة أو الإعداد لهذا البرنامج .

طرق الإعداد :

الإعداد والتشغيل لهذا البرنامج سوف نتناوله من خلال برنامج Windows XP باستخدام الطرق التالية :

أولاً : الإعداد من خلال الرمز My Computer

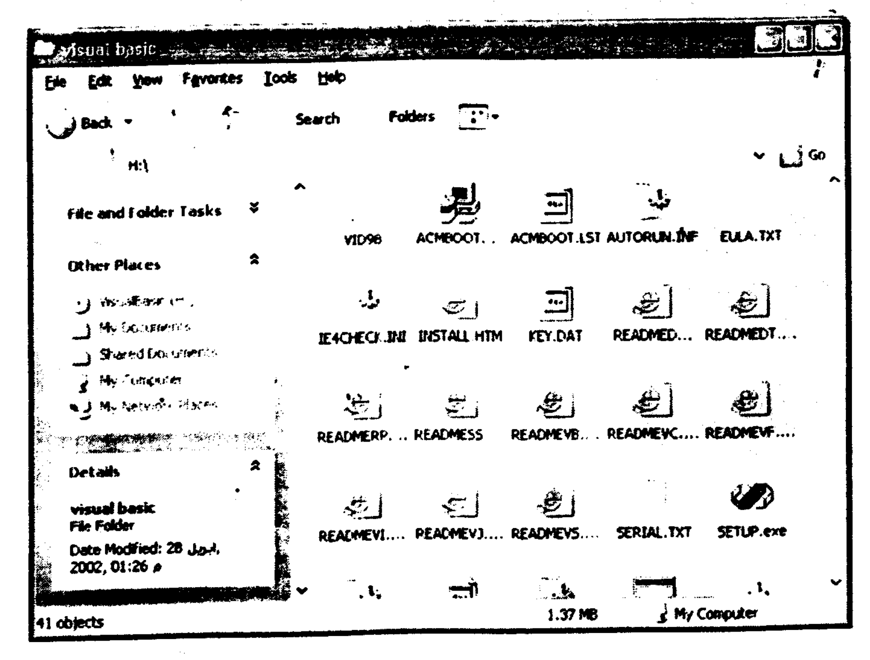
بالطبع هذا الرمز موجودة على سطح المكتب الخاص ببرنامج النوافذ ، وللتعامل معها يمكن تنشيطها وبالتالي فتحها، وسيظهر محتوياتها كما يظهر بالشكل التالي :



الإعداد من خلال الاسطوانة المضغوطة :

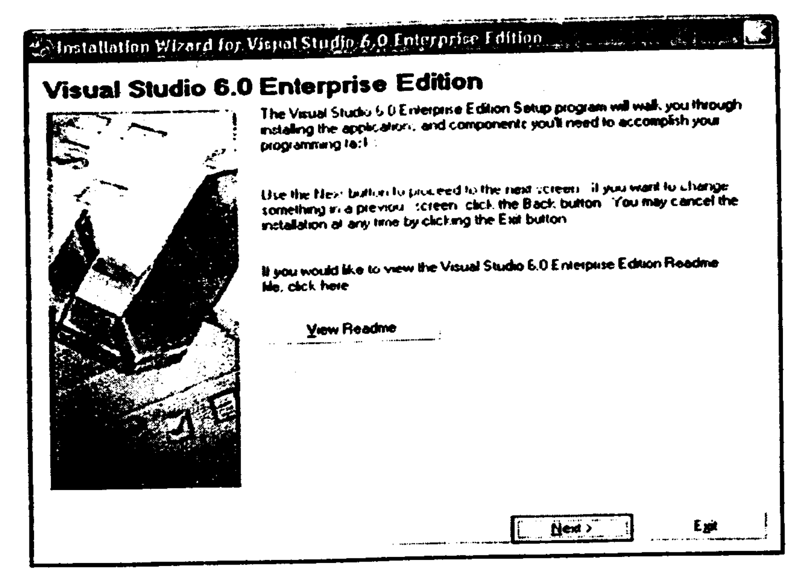
١- وضع الاسطوانة بمحرك الأقراص CD ROM الموجودة بالجهاز.

٢- سيتم التشغيل التلقائى للاسطوانة. أما فى حالة عدم التشغيل التلقائى للاسطوانة يتم فتح الرمز My Computer أو جهاز الكمبيوتر، وتحريك المؤشر إلى الرمز الخاص بوحدة إدارة أقراص الليزر ولتكن هنا على سبيل المثال تحمل الحرف (H) وفتحها، وبالتالي سيظهر محتويات هذه الاسطوانة من المجلدات أو الأدلة أو الملفات. كما سيظهر من الشكل التالى:

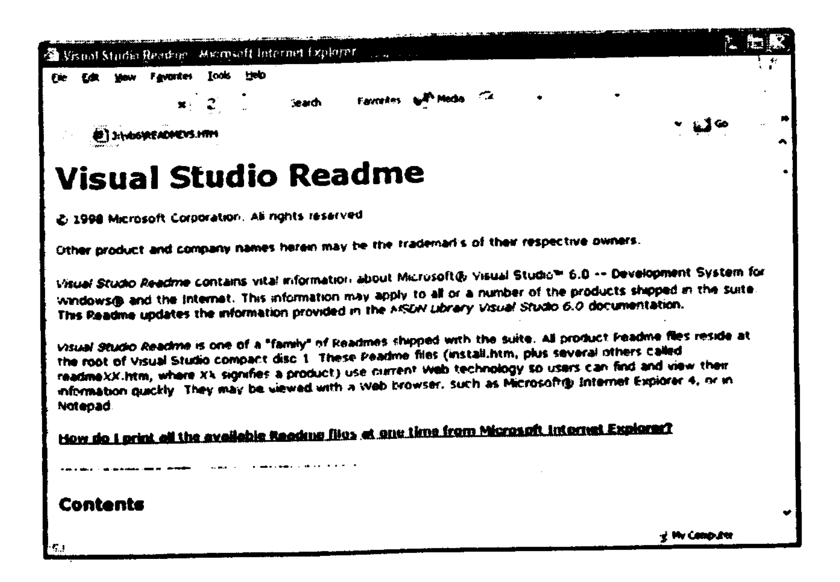


من هنا يمكن تنشيط النقر المزدوج على الأمر setup لتظهـــر نافـــذة بـــدء التشغيل الخاصة بالبرنامج.

ليظهر الشكل التالى: .

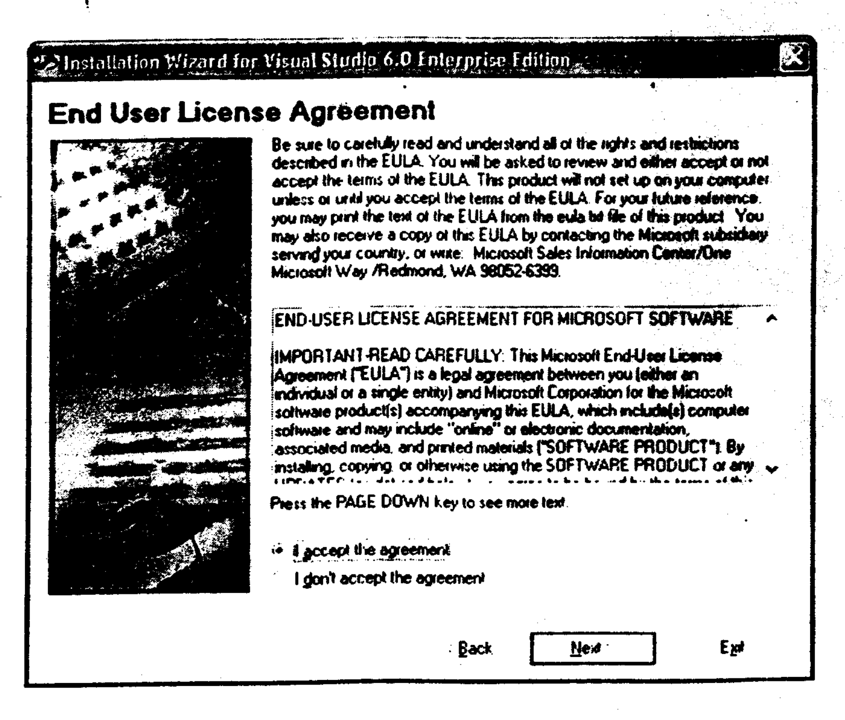


وعند النقر على الزر View Readme تظهر النافذة التالية:



ولمتابعة الإعداد يتم:

٣ – الضغط على الزر Next لتظهر النافذة التالية :



فيتم اختيار الموافقة على اتفاقية البرنامج ثم الضغط على Next لتظهر النافذة التالية:

Installation Wizard for Visual Studio 6.0 Professional Edition

Product Number and User ID

Please enter your product's ID number

Please enter your name and your company's name

Your name:

DrReffat

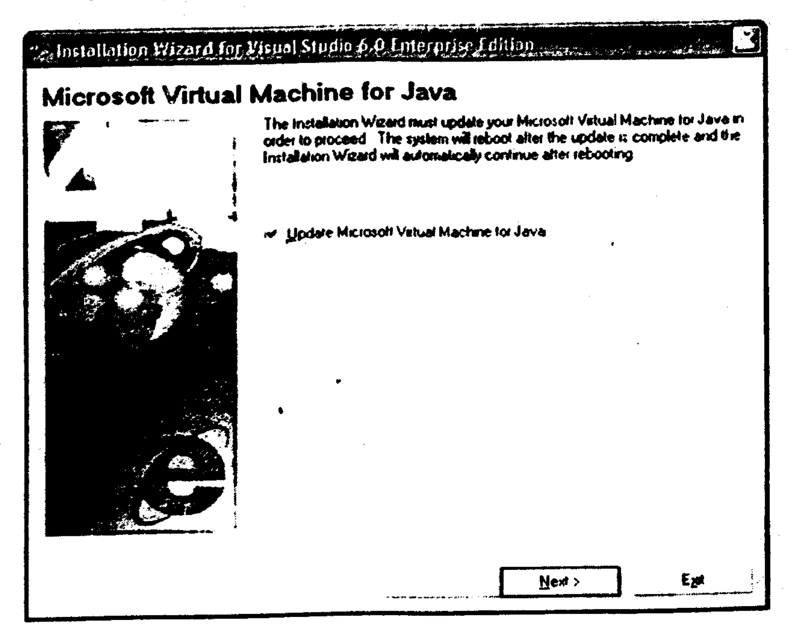
Your company's name:

ae

< Back Next > Exit

ويتمّ بها كتابة الرقم المسلسل للبرنامج والذى يأتى مع الاسطوانة عند شـرائها بالإضافة لاسم المستخدم وشركته.

ثم الضغط على Next للمتابعة، لتظهر النافذة التالية:



وهى خاصة بتحديث ملفات النظام بالجهاز فيتم التأكد من تحديد الاختيار ثم الضغط على Next، ليبدأ التحديث:

Microsoft VM for Java

The Microsoft VM for Java is now being installed on your computer

Installing Component: Register Base Classes

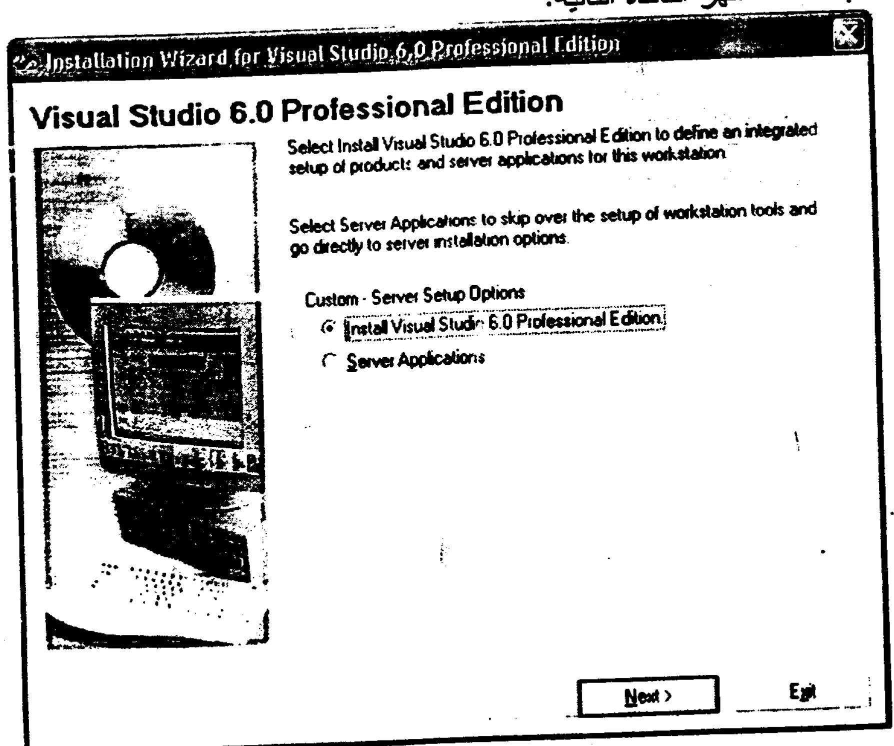
وربما يحتاج الأمر لإعادة تشغيل الجهاز وذلك فى حالة ظهور الرسالة التالية:

Installation Wizard for Visual Studio 6.0 Professional Ed...

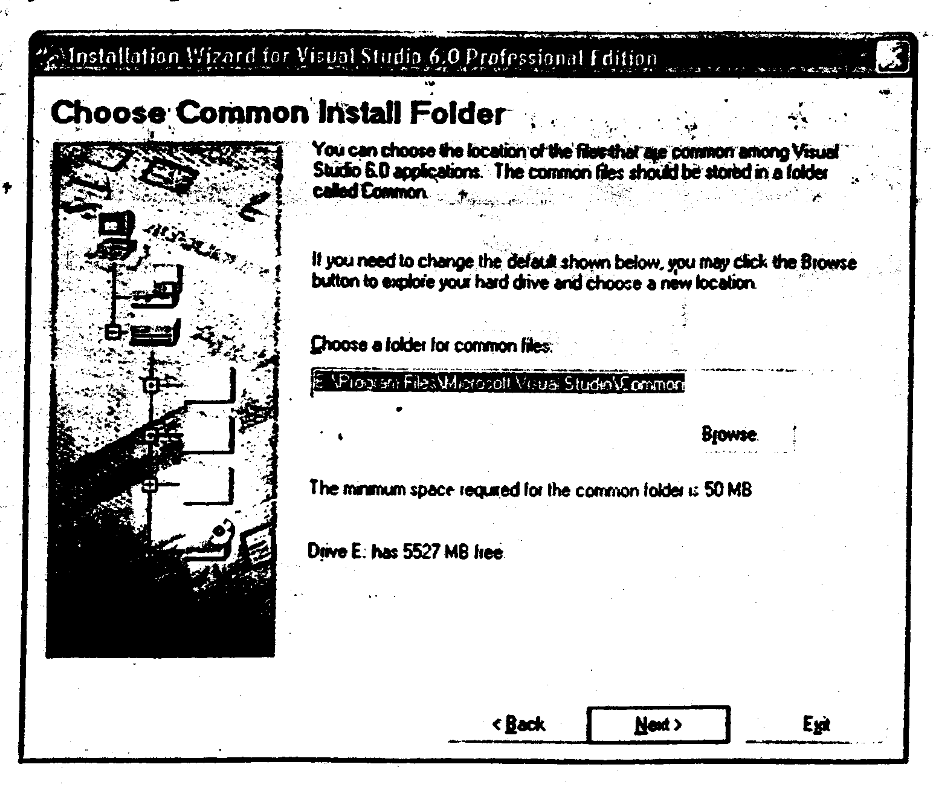
The Installation Wizard will now reboot the system.

OK

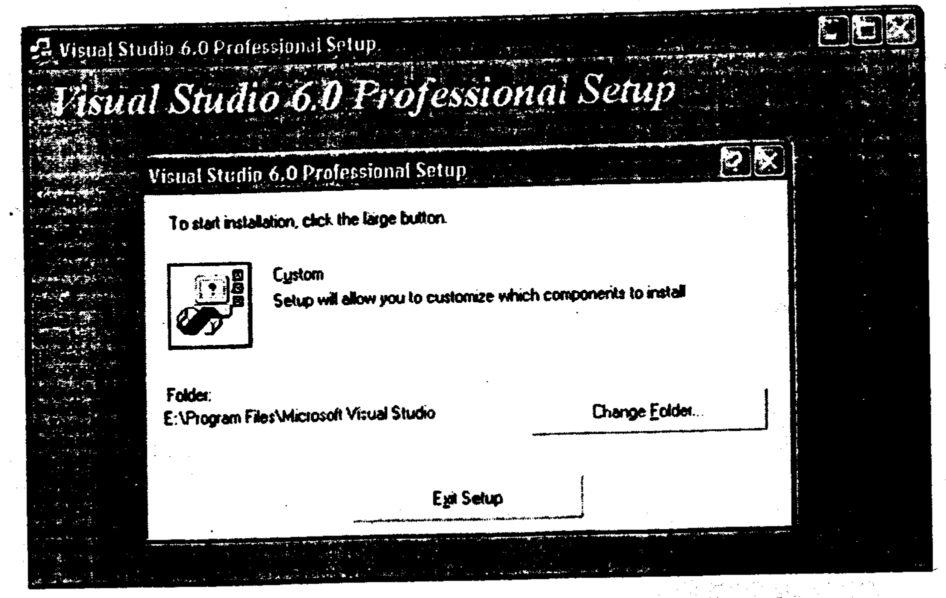
بعد ذلك تظهر النافذة التالية:



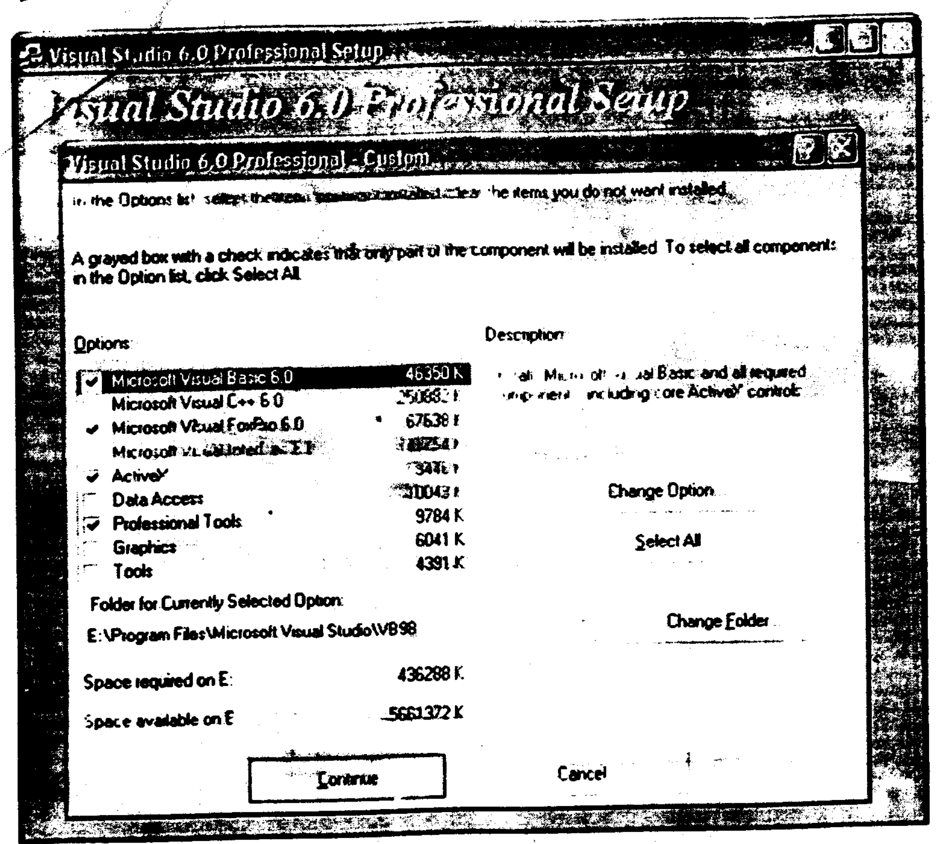
ومنها يترك الاختيار الافتراضي وهو الاختيار الأول، ثم يتم الضغط على الزر Next لتظهر النافذة التالية:



فيتم الضغط على Next لتظهر بعدها بعض الرسائل فيتم الضغط على Continue حتى تظهر النافذة التالية:



ومنها يتم الضغط على الزر Custom لمتابعة إعداد البرنامج علـــى القـــرص الصلب من خلال عرض قائمة بكامل البرامج الملحقة باســـطوانة Visual Studio Professional فيتم التأكد من اختيار الاختيار الأول فيها وهو : Microsoft Visual Basic 6.0 انظر الشكل التالي:



ثم يتم الضغط على الزر Continue للمتابعة، بعد ذلك يبدأ الإعداد في الاستمرار تلقائيا حتى ينتهي:

Microsoft®
Visual
Studio



Visual Studio™ 6.0 Professional Edition is the complete suite for rapidly building business solutions that unleash the power of Windows® and the Web

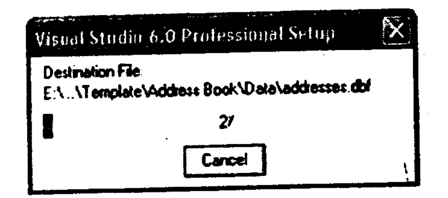
Features Include
A complete Web development environment

End-to-end debugging support.

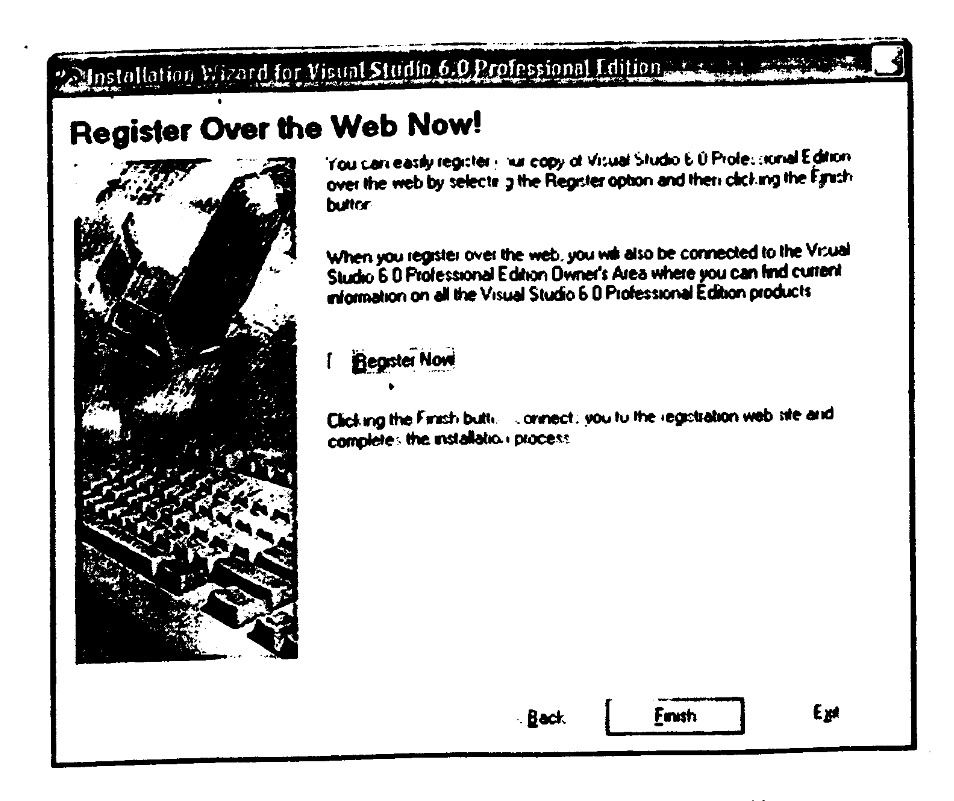
Integrated Professional Visual Database Tools

Productivity-boosting tools like IntelliSense technology

A complete development suite for Windows



وهكذا يتم متابعة جميع الرسائل الخاصة بالإعداد حتى ظهور رسالة تفيد بالانتهاء من الإعداد، والمطالبة بإعادة تشغيل الجهاز، وبعد إعادة التشغيل تظهر الرسالة التالية:



ويتم إزالة التحديد من الاختيار Register Now ثم الضغط على الزر Finish لإنهاء الإعداد؛ عند ذلك يمكنك تشغيل البرنامج والبدء فى استخدامه.

ثانياً : الإعداد من خلال أمر تشغيل RUN الموجود بقائمة ابدأ Start :

بفرض أن وحدة إدارة الاسطوانات المضغوطة هي .H ويوجد بها حاليا اسطوانة البرنامج وبالتالي يجب إتباع الآتي لتنفيذ عملية التهيئة من خلال CDROM :

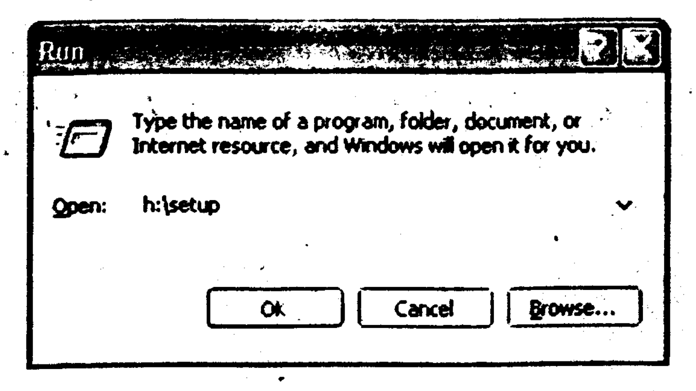
١ - النقر على زر أو أمر ابدأ Start ، ظهور القائمة الخاصة به .
٢ - النقر على أمر تشغيل RUN ، عندئذ سيظهر مربع حواري .
٣ - كتابة أمر الإعداد التالي (في حالة معرفة اسم وحدة الإدارة والأدلة الفرعية المراد التعامل معها واسم الملف أو البرنامج التنفيذي المراد تشغيله) ،

H:\setup

ثم الضغط على مفتاح Enter أو النقر على امر موافق بعد تحريك المؤشر عليه.

ملاحظة :

في حالة عدم التأكد من معرفة اسم وحدة الإدارة أو الأدلة الفرعية المراد التعامل معها أو اسم الملف أو البرنامج التنفيذي المراد تشغيله) ، يمكن تنفيذ الأمر من خلال أمر استعراض Browse الموجود في المربع الحواري الخاص بأمر تشغيل RUN ، وبالتالي يمكن التعامل مع مكوناته كل على حده للوصول إلى البرنامج التنفيذي Setup وفيما يلى نافذة (المربع الحوارى) لأمر التشغيل Run .

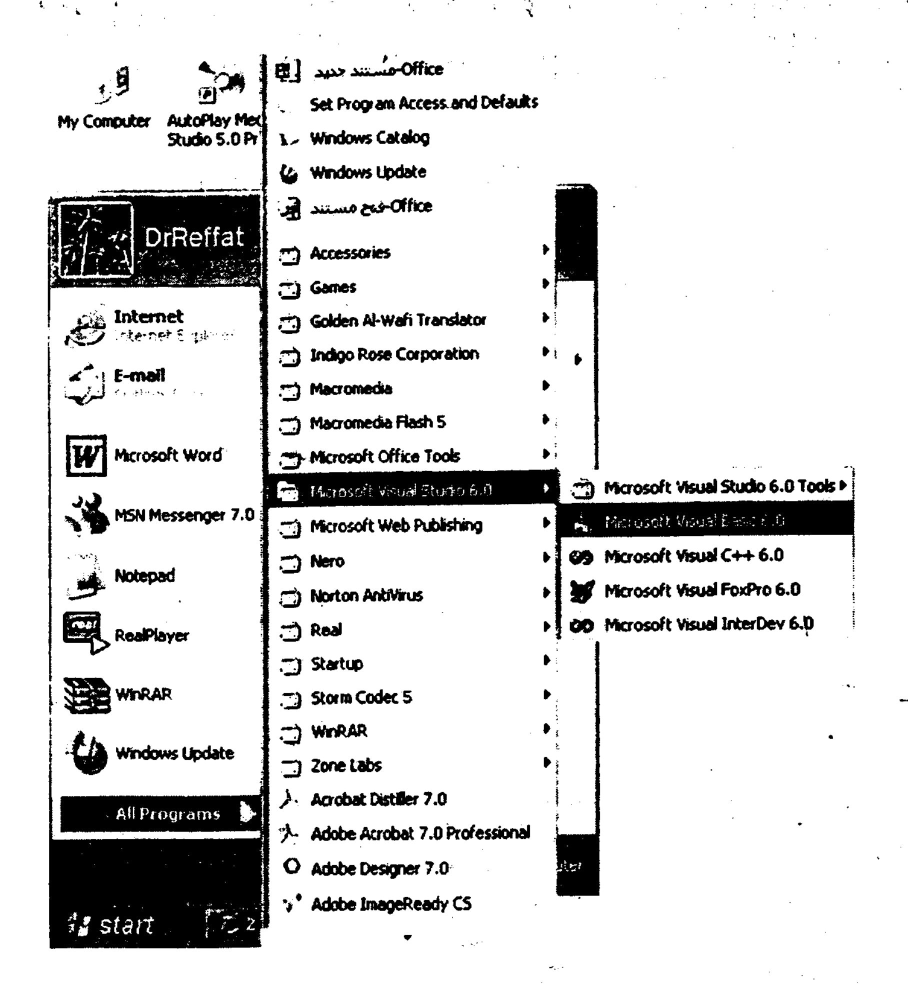


١ - ٣ كيفية تشغيل Microsoft Visual Basic 6.0 :

التشغيل من خلال النوافذ Windows XP :

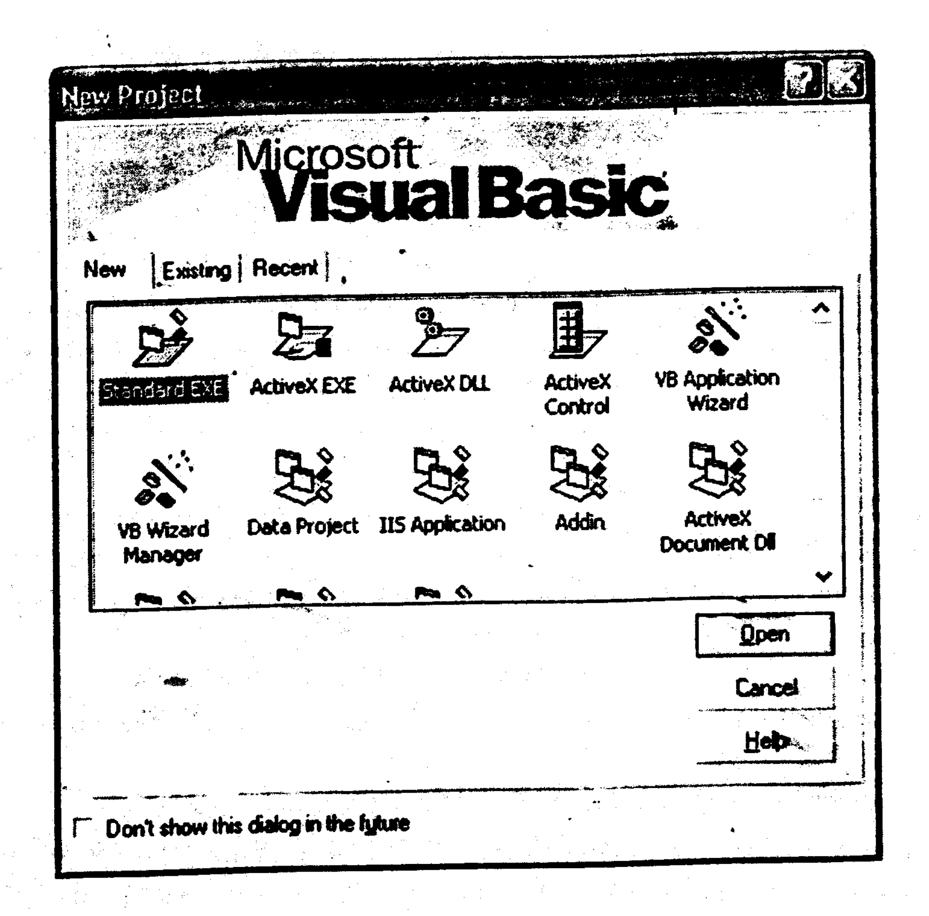
بعد الانتهاء من الإعداد السابق يتم إلحاق رموز تشغيل البرنامج بقائمة ابدأ Start وبالتحديد داخل البرامج All Programs، وبالتالى لتشغيل هذا البرنامج يمكن تنفيذ الآتي :

— فتح قائمة ابدأ Start ثم الذهاب للقائمة الفرعية All Programs والوقوف عليها لفتحها، أنظر الشكل التالى:



- ثم الذهاب إلى الأمر القائمة الفرعية Microsoft Visual Studio 6.0 والوقوف عليها أيضا لفتحها ثم بعد ذلك تظهر قائمتها الفرعية فيتم اختيار منها البرنامج

Microsoft Visual Basic 6.0 بالضغط بالزر الأيسر للفارة لتشغيله، لتظهر النافذة التالية:



وبالتالى يمكن التعامل مع نوافذ وقوائم وأوامر هذا البرنامج كما سيتضح ذلك فيما بعد .

١- ٤ مراحل إعداد البرنامج المصدري :

أ- من خلال Microsoft Visual Basic 6.0 :

المقصود من ذلك كتابة برنامج المستخدم من خلال برنامج Visual Basic، بالطبع البرمجة هنا تختلف عن البرمجة فى اللغات العادية مثـل مترجمـات لغـة البيسيك العادية BASIC سواء المُستخدم على الأجهزة الشخصية أو الأخرى .

كما نعرف أن مترجمات لغة البيسيك منها ما هـو تفسـيرى , BASIC) BASICA , GW-BASIC ..) وما هو تجميعى مثل (QB , Gwbasic ...) وللتعامل معها يجب معرفة أوامرها المستخدمة فى المترجم الموجود على الجهـاز الذى سيتم تنفيذ البرنامج عليه، ثم بناء البرنامج بالكامل من خلال كتابـة الأوامـر والتعليمات المختلفة بيد معد البرنامج فى برنامجه الذى يبنيه، معنى ذلك أن معـد البرنامج يقوم بتصميم البرنامج بالكامل من خلال الأوامر التى يكتبها.

هنا بالطبع سيجد معد البرنامج خلال مراحل بناء برنامجه كثير من المشاكل أثناء عملية الإعداد له، منها على سبيل المثال فقد كثير مـن الوقـت والمجهـود والتفكير فى حل المشاكل - الشكل العام للبرنامج أو واجهة البرنامج ليست جميلـة كما يجب لقصور إمكانيات اللغة - انعدام أو ضعف المرونة فـى التعامـل مـع البرنامج والتنقل داخله وتشغيل وعرض عناصره - الإمكانيات المحدودة للبرنامج من حدود الملفات المختلفة وصعوبة التعامل معها بصورة جيدة أو سهلة ... الخ .

ولكن استخدام برنامج البيسيك المرئى Visual Basic فى عمليات البرمجة فعلا قد سهّل وحل كل أو أغلب المشاكل الموجودة فى البيسيك العادى، بل حسّـن الأداء وواجهة البرنامج وسهّل على المستخدم التعامل مع برامج التطبيقـات التـى يعدها مخططى البرامج العادين لحل المشاكل العادية أو اليومية.

** ولإعداد أو كتابة برامج المستخدم بواسطة برنامج Visual Basic يمكـن إتباع الآتي :

١ - التعرف على المشكلة المراد حلها آليا :

أى مراعاة المتبع فى البرمجة العادية من خلال اللغات أو برامج التطبيقات الجاهزة، من حيث :

- تحديد الهدف من حل المشكلة .
- تحديد عناصر المخرجات المطلوب الوصول إليها .
- تحديد عناصر المدخلات الخاصة ببيانات البرنامج .
- تحديد أسلوب وطرق الوصول لحل المشكلة، وبالطبع التعرف على تفصيلات العناصر السابقة ... الخ ، وبالتالى وضوح فكرة البرنـــامج بالكامــل لمعــد أو مخطط البرنامج قبل البدء فى كتابة البرنامج .
- وبالتالى يمكن التعرف على النقاط الغير واضحة الحل أو الصعبة .
- ثم التفكير فى حلها .
- ثم البدء فى تقسيم البرنامج إلى أجزاء مستقلة، حيث يقوم كل جزء منها بـــأداء دور ووظيفة محددة .
- ثم البدء فى المرحلة الرئيسية التالية (رقم ٢) .

٢ - تصميم واجهة البرنامج Drawing The interface :

يتم تنفيذ ذلك بعد معرفة المرحلة السابقة بالكامل .

- وبالتالى يجب تحديد عدد النوافذ التى يحتاجها البرنامج ورسـمها أولاً علـــى المستندات الورقية بالشكل المطلوب تصميمه .
- ثم تحديد ما هى الأدوات التى يمكن أن يحتاجها هذا البرنامج.

- وبالتالى البدء فى تصميم شاشات واجهة البرنامج سواء القوائم الرئيسية أو الفرعية أو شاشات الإدخال أو الإخراج بناء على عناصر الإدخال والإخراج الخاصة بالمشكلة محل الدراسة على نافذة البرنامج Form.

على سبيل المثال:

تصميم شاشة القائمة الرئيسية للبرنامج، حيث تحتوى مثلاً على مجموعة من الأقسام مكتوبة باللغة العربية أو الإنجليزية، تحتوى على مراحل تشغيل البرنامج (التطبيق) ومنها مرحلة إضافة البيانات - مرحلة التعديل فى البيانات - مرحلة تشغيل البيانات - مرحلة الصيانة للبيانات - مرحلة إعداد التقارير - إنهاء البرنامج ... الخ من العمليات التى يحتاجها مصمم البرنامج فى تطبيقه.

وبالطبع كل مرحلة من هذه المراحل سيقابلها تنفيذ برنامج فرعى أو أكثر من خلال نافذة الكود Code وبالطبع ربطها بالنوافذ الأخرى الموجودة فى برنامج VB.

٣ - ضبط الخصائص Setting Properties :

تتم هذه المرحلة بعد وضع أو تصميم الأدوات المراد استخدامها فى البرنامج على نافذة البرنامج Form، وذلك لتحديد سلوك ووظائف كل أداة فى البرنامج، وتوجد هذه الخصائص فى نافذة الخصائص الملحقة ببرنامج البيسيك المرئى Visual Basic.

حيث توجد لكل أداة من هذه الأدوات مجموعة من الخصائص Properties تختلف من واحدة إلى أخرى، حيث تحدد شكلها ولونها والخط المستخدم فيها وعنوانها ... الخ .

وبالتالى يتم تغيير الخصائص الافتراضية الممنوحة لهذه الأدوات من برنامج VB لتناسب برنامجك.

على سبيل المثال :

بفرض أن خلفية نافذة البرنامج التى يمنحها نظام النوافذ Windows من خلال لوحة التحكم المسماة Control Panel قد تكون زرقاء وبالتالى سيرتبط ذلك اللون بالخلفية فى نافذة برنامج VB وفى حالة الحاجة لتغيير هذا اللون إلى أى لون أخر فى نافذة برنامجك فى VB وبالتحديد فى نافذة الشكل Form ، ما عليك إلا تحديد اللون من خلال خاصية لون الخلفية المسمى Back Color للنافذة الحالية.

٤ - كتابة التعليمات Writing Code :

بعد الانتهاء من المراحل السابقة جميعها بالطبع تكون الصورة أو المشكلة محل الدراسة واضحة أمام مخطط البرنامج وبالطبع يجب عليه ترجمة هذه الصورة إلى ما يناظرها من أوامر وتعليمات ووظائف فى صورة مجموعة من البرامج الفرعية المرتبطة بالبرنامج الرئيسى الخاص بتشغيلها.

ويتم ذلك من خلال نافذة الكود Code الملحقة بنوافذ برنامج VB وبالطبع تختلف تعليمات كل برنامج عن الأخر من حيث نوعها وحجمها ...الخ.

ويتم كتابة هذه التعليمات فى مواضع معينة فى VB وهى ثلاثة أماكن كما يلى:

١ - الإجراء Procedure :

بالطبع هو المكان أو الوعاء الذى يحتوى على التعليمات المختلفة والذي يحتفظ بها ، ثم يمكن تنفيذها من خلال (البرنامج).

والكتابة هنا يمكن أن تكون فى موضعين مختلفين هما :

أ - إجراء الحدث Event Procedure :

وأوامر هذا البرنامج ترتبط بحدث معين (راجع جزء الأحداث) حيث يتم تنفيذه من خلال حدث يتم بواسطة المستخدم مثل:

Click , Keypress أو حدث بواسطة برنامج مثل Load , Unload , Chang . مع العلم لا يمكن الوصول إلى هذا الإجراء Procedure إلا من خلال الحدث المحدد له، ويتم تنفيذ ذلك من خلال النقر المزدوج على الحدث نفسه أثناء مرحلة التصميم للبرنامج .

ب - برنامج نافذة البرنامج Procedure Of Form :

حيث يتم تنفيذ هذا البرنامج من خلال أى منطقة فى نافذة البرنامج Form، يمكن أن يستدعى من خلال اسمه أو يمكن الوصول إليه من خلال أى إجراء فى نافذة البرنامج Form أو قد يستدعى من خلال الأمر Call .

٢ - إجراء المشروع الرئيسى Procedure Of Project :

وهو الإجراء الذى يكتب من خلال Module ، ويمكن أن يتم تنفيذه من خلال أى نافذة برنامج Form وأيضاً يمكن استدعاؤه من خلال أمر Call .

٣ - الدوال Function

هو إجراء خاص بالدوال Function، ويمكن أن يستدعى من خلال كتابة اسمه فقط.

وسوف نعرض فى هذا الكتاب نماذج لهذه البرامج.

✍ ملاحظة :

الإجراء قد يكون برنامج فرعى خاص بأداء وظيفية محددة قد تحتاج إلى تنفيذها فى البرنامج فى مواد معينة ، ولذا بدلاً من تكرار هذه الأوامر فى أكثر من موضع فى البرنامج لأداء الوظيفة التى تريدها ، فإنه يمكن كتابة الأوامر الخاصة بهذا الإجراء وحفظها تحت اسم معين ، مع العلم بأن الإجراء يبدأ بكلمة ***sub*** ثم يليها اسم الإجراء الذى ستستخدمه أنت فى المناداة عليه فى برنامجك، ثم يلى ذلك فى السطر التالى التعليمات والأوامر التى تريدها، ثم ينتهي الإجراء بكلمتي ***End Sub***.

٥ - ترجمة البرنامج إلى لغة الآلة :

بالطبع كتابة التعليمات السابقة تتم فى الشكل المصدرى (اللغة الإنجليزية العادية) ولتنفيذها لا تتعرف عليها الآلة، وبالتالى لابد من ترجمة هذه التعليمات إلى ما يناظرها من أوامر مكتوبة بلغة الآلة حتى يستطيع الحاسب التعرف عليها وتنفيذها مباشرة وبسهولة .

والأوامر التى يتم كتابتها فى المرحلة السابقة تحفظ فى ملف برنامج واحد أو أكثر من ملف برنامج فى الشكل المصدرى، وعند الترجمة له سيتم وضع كل هذه البرامج (الملفات) فى برنامج واحد يأخذ امتداد الملفات التنفيذية EXE. أى Executable file ، أى يمكن تنفيذه مباشرة من خلال علامة نظام التشغيل بدون الحاجة إلى البرامج المصدرية (الأصلية) .
وتمتاز هذه النوعية من الملفات بكثير من المميزات منها السرعة فى التنفيذ، الأمان والسرية لمحتويات البرنامج حيث يكون فى شكل لغة الآلة، ملف برنامج واحد فقط يتولى تنفيذ كل البرامج الفرعية له ...الخ .

كيفية الترجمة :

يتم ذلك من خلال القائمة الفرعية الأولى ملف File باختيار أمر الترجمة التنفيذية Make Project1 .exe ، وبتنفيذ هذا الأمر ستظهر نافذة تطلب من المستخدم كتابة بعض عناصر بدء الترجمة مثل اسم البرنامج التنفيذى ...الخ وبعد الضغط على زر التنفيذ سينتج ملف برنامج تنفيذى بنوعية EXE.

Make Project

Save in: VB98

Template
Tsql
Wizards
C2.EXE
CVPACK.EXE
LINK.EXE
VB6.EXE
VISDATA.EXE
الترحيب.exe

File name: Project1.exe

OK
Cancel
Help
Options...

- بالتالي يتم كتابة اسم البرنامج في الصندوق المقابل File name ، ثم الضغط علي OK لتنفيذ ذلك.

١ - ٥ أنواع الملفات المستخدمة في برنامج البيسيك المرئي VB :

يمكن من خلال البيسيك المرئى التعامل مع مجموعة من الملفات وذلك لتنفيذ البرنامج أو المشروع أو التخطيط المطلوب إعداده وأهمها ما يلى :

١ - ملفات المشروع Project

نوعية أو (امتداد الملف) : EXE

هو الملف الرئيسي لهذا النظام، الذى يحتوى على برنامج المشروع بالكامل، أى يحتوى على جميع الملفات الأخرى التى يمكن أن تلحق بملف المشروع مثل ملف نافذة البرنامج Form وملف النموذج أو الكود Modules .. الخ . ويتم تنفيذه من خلال أمرى Save Project أو Save Project As من خلال القائمة الفرعية الأولى File .

٢ - ملفات نافذة البرنامج Form

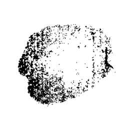
- نوعية أو (امتداد) الملف : FRM

هذه النوعية خاصة بنافذة البرنامج المسماة Form ، وهى التى تستخدم لتصميم شاشات الإدخال أو الإخراج لبيانات البرنامج أو شاشات القوائم المستخدمة فى تشغيل البرنامج .

ويتم تنفيذها من خلال أمر Form الموجود فى قائمة المشاهدة View أو من خلال الضغط على مفتاحى Shift + F7 .

يمكن أن تعطى افتراضيا عند تنفيذ أمر فتح مشروع جديد New Project من قائمة File .





مع العلم بأن النظام يعطى لها اسم افتراض من عنـده اسـمه Form1 ، ويمكن للمستخدم للبرنامج من تغيير هذا الاسم من خلال نافذة الخصائص وبالتحديد من أمر Caption بكتابة الاسم الذي يريده .

ويلاحظ أيضاً أنه يمكن أن يتم تصميم مجموعة من نافذة البرنامج Form فى البرنامج الواحد .

٣ - ملفات البرنامج (التعليمات) Modules

- نوعية أو (امتداد) الملف : BAS

هذه النوعية من الملفات خاصة ببرنامج البيسيك نفسه الـذى يـتم تنفيـذه وربطة مع نافذة البرنامج Form أو مع النوافذ الأخرى .
ويتم التعرف عليه من التطبيق وفى الغالب يطلق عليه Code Module.

وهذا الملف بالطبع يحتوى على جميع خطوات حل المشكلة التى بصـددها البرنامج ، أى يحتوى على جميع التعليمات الخاصة بالمشكلة محل الدراسـة فـى صورة أوامر لغة VB .

٤ - ملفات أدوات التحكم Custum Controls

- نوعية (امتداد) الملف : DLL
هذا الملف مرتبط بنافذة الأدوات الخاصة ببرنامج Visual Basic. ويمكن للنظام أن يقوم بزيادة هذه الأدوات فى صندوق Tool Box .

٥ - ملفات النوافذ المتعددة المستندات Multiple Documents Interface Forms

- نوعية (امتداد) الملف : MDI

تتكون هذه النوعية من أكثر من نافذة برنامج Form ، حيث يمكن أن يضاف إليها هذه النوافذ داخل المشروع الواحد، ولا يمكن أن يكون فى المشروع الواحد أكثر من ملف MDI .

٦ - ملفات النصوص Text File

- نوعية (امتداد) الملف : TXT

هذا الملف خاص بملفات النصوص (البيانات الحرفية أو الرقمية) التى يمكن أن تلحق بملف البرنامج (المشروع) لإظهارها والتعامل معها، ويحتوى هذا الملف على البيانات فى صورة ASCII Text .

يتم تنفيذ أو إنشاء هذا الملف من نافذة File وبالتحديد عند حفظ الملف من خلال أمر حفظ باسم Save As ، ثم اختيار نوعية الملفات TXT .

✍ ملاحظة :

بالطبع هناك كثير من الملفات الأخرى التي لم نتطرق إليها لعدم احتياجها في هذه المرحلة.

١ - ٦ الكتابة فى برنامج البيسيك المرئى VB

يمكن الكتابة في برنامج البيسيك المرئي فى النوافذ أو فى الملفات باستخدام اللغة العربية أو اللغة الإنجليزية وذلك بإتباع الآتي :

أولا : الوضع الافتراض للبرنامج :

الذى يمنحه برنامج البيسيك المرئى هو استخدام اللغة الإنجليزية .
وبالتالى عند الكتابة على نافذة البرنامج Form لبعض الأدوات المستخدمة عليه، فإنه سيتم كتابها باللغة الإنجليزية. وأيضاً عند التعامل مع النوافذ الأخرى التى يمكن الكتابة فيها مثل نافذة الخصائص أو الكود فإنه فى هذا الوضع سيتم الكتابة باللغة الافتراضية الإنجليزية، وهذا هو الوضع العادى عند تشغيل البرنامج .

ثانياً : التغيير إلى اللغة العربية :

يمكن أن يتم تغيير اللغة المستخدمة عند الكتابة على نافذة البرنامج Form على سبيل المثال لبعض الأدوات المستخدمة عليه وذلك باللغة العربية وذلك من خلال الآتي :

- نقل الأداة إلى نافذة البرنامج التى يمكن الكتابة فيها مثل أداة نص ثابت Label (أداة العنوان A) أو أداة نص متغير Text Box (صندوق النصوص ab)، وظهور الصندوق الخاص بها والذى يمكن استخدامه فى الكتابة فيه .
- هنا لتغيير اللغة إلى عربى يمكن الضغط على مفتاحى Alt + Right Shift .
- ثم يتم الكتابة للكلمة بالأحرف العربية، وسيظهر ذلك فى صندوق الدالة ، (باستخدام طريقة الزحزحة للأحرف فى اتجاه اليمين) أي الكتابة بطريقة عكسية للأحرف .

- ثم الضغط على مفتاح Enter أو الضغط على الزر الأيسر لمؤشر الفارة.
- وفى حالة العودة إلى الأصل والكتابة باللغة الإنجليزية يمكن تغيير ذلك من خلال الضغط على مفتاحى Alt + Shift Left ، وبالتالى يمكن الكتابة باللغة الإنجليزية وهكذا .

✍ ملاحظة :

أنه يمكن التعديل فى النص المدخل على النافذة من خلال نافذة الخصائص Properties النشطة مع البرنامج أثناء تشغيله وبالتحديد من خلال أمر Caption فى الجزء أو الصندوق المجاور له، وذلك بتحريك المؤشر الظاهر بهذا الصندوق إلى مكان الخطأ أو المراد تعديله والكتابة عليه بعد تحديده أو إلغاء الجزء المحدد باستخدام مفتاح Del من لوحة المفاتيح.

وأيضاً عند التعامل مع النوافذ الأخرى التى يمكن الكتابة فيها مثل نافذة الخصائص أو الكود فإنه فى هذا الوضع سيتم الكتابة باللغة الافتراضية الإنجليزية، وهذا هو الوضع العادى عند تشغيل البرنامج .

١ - ٧ عناصر لغة البيسيك المرئي (المتغيرات والثوابت والحقول ....)

بالإضافة إلى المكونات الخاصة باللغة السابق التعرف عليها مثل الملفات والبرنامج، يمكن التحدث عن المكونات الفرعية أو التفصيلية لهذه الملفات المختلفة على أساس أن اللغة بصفة عامة تتكون من خلال ترابط مجموعة من العناصر المكونة لها، ومن هذه العناصر (حروف اللغة والمتغيرات والثوابت والحقول ...).

أولاً : الحروف المختلفة المستخدمة فيها :

تتكون من العناصر التالية :

أ - الحروف الأبجدية	Alphabetic
ب - الأرقام	Digits
ج - الحروف الخاصة	Special Character
د - العوامل الحسابية	Arithmetic Operation
- الأس	Exponentiation
- علامة الضرب	Multiplication
- علامة القسمة	Division
- علامة الجمع	Addition
- علامة الطرح	Subtraction

هـ - عوامل العلاقات (المقارنة)

التساوى =	Equal To
أكبر من >	Greater Than
أصغر من <	Less Than
لا يساوى <>	Not Equal to
أكبر من أو يساوى >=	Greater than Or Equal
أصغر من أو يساوى <=	Less Than Or Equal

و - رموز الفواصل :

النقطة .	Decimal Point
الفصلة ,	Comma
الفصلة المنقوطة	Semicolon
النقتطان	Colon
علامة التنصيص	Quotation Mark
قوس اليسار	Left Parenthesis
قوس لليمين	Right Parenthesis
علامة الاستفهام	Question Mark
علامة الإضافة - و	Ampersand &
علامة النسبة المئوية	Percentage
علامة الدولار	Dollar Sign
علامة العدد	Number Sign

ثانياً : المتغيرات Variables

* **تعريف المتغيرات :**

هى أسماء لمواضع تخزين البيانات فى ذاكرة الحاسب والمرتبطة ببرنامج المستخدم، حيث قد تتغير محتويات هذه المتغيرات أثناء تشغيل البرنامج (مرحلة التشغيل) .

* **الإعلان عن المتغيرات :**

الإعلان عن المتغير فى برنامج VB معناه أن تخبر برنامج VB عن شيئين هامين هما (اسم المتغير، ونوعه).

ونظراً لاختلاف البرنامج أو طبيعة البرنامج من كونه برنامج إجراء خاص بحدث معين أو نافذة برنامج أو إجراء مشروع، فإن الإعلان عن المتغيران فى كل من هذه المواضع قد يختلف نظراً للموقع.

على سبيل المثال :

أ - الإعلان عن المتغيرات فى الإجراء الخاص بالإجراء العام الخاص بالـ Project:

فى هذا البرنامج يكون المتغير متاح الاستخدام لكل برامج النماذج الأخرى التابعة لهذا المشروع أو التطبيق،
ويتم تنفيذ ذلك من خلال تعريف المتغير من خلال الأمر Global .

ب - الإعلان عن المتغيرات فى الإجراء الخاص بنافذة البرنامج Form:

فى هذا البرنامج يكون المتغير متاح الاستخدام داخل الإجراء الخاص بنافذة البرنامج فقط Form ، ويتم تنفيذ ذلك من خلال كتابة الأوامر Dim أو Static .

ج - الإعلان عن المتغيرات فى الإجراء الخاص بالحدث Event :

فى هذا البرنامج يكون المتغير متاح الاستخدام داخل الإجراء الخاص بالحدث Event فقط، ويتم تنفيذ ذلك أيضاً من خلال كتابة الأوامر Dim أو Static .

ملحوظة :

الإعلان عن المتغيرات قد يتم بإحدى الأوامر التالية :

Dim أو Global أو Static

ولكن يهمنا أن نعرف أنه فى حالة الاحتفاظ بقيمة المتغير الخاص بإجراء معين لاستخدامها داخل المشروع بصفة عامة، يجب أن يستخدم

الأمر static لتعريف المتغير على أنه عام الاستخدام على مستوى الإجراء الحالى، معنى ذلك أن المتغيرات لها مدى معين يتم تحديدة بناء على العنوان الذى تم تحديده له.

*** أنواع المتغيرات :**

✍ ملاحظة : (يجب ملاحظة أن تاتى كلمة AS قبل نوع المتغير).

١ - المتغيرات الحرفية String

هى مواضع تخزين بيانات فى الشكل الحرفى أو النصي، أى لا يمكن أن تستخدم فى العمليات الحسابية المختلفة.

٢ - المتغيرات العددية الصحيحة Integer

هى مواضع تخزين البيانات فى الشكل العددى الصحيح الذى لا يحتوى على علامة عشرية.

٣ - المتغيرات العددية الصحيحة الطويلة Long

هى مواضع تخزين البيانات فى الشكل العددى الصحيح الذى لا يحتوى على علامة عشرية ولكن مساحته أو مداه مضاعفة عن المتغير السابق.

٤ - المتغيرات الحقيقية المفردة Single

هى مواضع تخزين البيانات فى الشكل العددى الحقيقى الذى يحتوى على علامة عشرية عائمة.

٥ - المتغيرات الحقيقية المزدوجة **Double**

هى مواضع تخزين البيانات فى الشكل العددى الحقيقى الذى يحتوى على علامة عشرية عائمة، ولكن مساحته أو مداه مضاعفة عن المتغير السابق.

٦ - المتغيرات الحقيقية العملة **Currency**

هى مواضع تخزين البيانات فى الشكل العددى الحقيقى الذى يحتوى على علامة عشرية ثابتة **Fixed Point**.

٧ - المتغيرات المتنوعة **Variant**

هذه النوعية من المتغيرات تتغير خاصيته الحرفية أو العددية (أى كانت) حسب محتوياته، أى يمكن أن يحتوى هذا المتغير على قيم حرفية أو عددية أو قيم خاصة بتاريخ.

✍ ملاحظة :

إذا لم تحدد نوعية المتغير فى البرنامج فسيصبح من النوع Variant

أمثلة :

- **متغيرات حرفية :**

```
var1 = "45"
var2 = " Damietta "
```

- **متغيرات عددية :**

```
var_sal = 200
```

✍ ملاحظات :

- خاصة بتحديد أو اختيار اسم المتغير :
- يجب أن يبدأ اسم المتغير بحرف هجائي .

- يجب ألا يزيد طول اسم المتغير عن ٤٠ حرفا .
- يجب عدم استخدام الكلمات المحجوزة الخاصة بالنظام كاسم للمتغير.
- يجب عدم استخدام أسماء الأدوات كاسم للمتغير.

ثالثا : الثوابت

نقال عنها الثوابت العددية وهى القيمة العددية التى لا تتغير قيمتهـا أثنـاء تشغيل البرنامج.

وهناك نوعين هما :

أ - الثوابت الصحيحة Integer Constants

هى القيمة العددية الصحيحة التى لا تحتوى على علامة عشرية.

ب - الثوابت الحقيقية Real Constants

هى القيمة العددية التي تحتوى على علامة عشرية (أى القيمة يوجد بهـا جزء صحيح وجزء كسرى) .

مثال :

```
For Counter = 1 to 10
      Print ......
      -----------
Next
```

مثال :

```
Const Number =  10
```

قد يوضع هذا الثابت فى بداية البرنامج أو الإجراء مره واحدة، وبالتالى يمكن استخدامه فى برنامجك فى أى وقت.

مثال :

```
Const Number =  10
For Counter = 1 to number
        ------
        ------
Next Counter
```

مثال :

```
Const Password = "dr_refat"
```

شرح الأمر :

حفظ كلمة dr_refat فى ثابت يسمى Password.

ملاحظة :

قواعد تحديد الإعلان الخاصة بالمتغيرات يتم تطبيقها بنفس النظام على الثوابت.

على سبيل المثال :

```
Global Const pass1="4321"
```

الفصل الثاني

التعامل قوائم

برنامج Visual Basic Ver 6.0

يتناول هذا الفصل بعض من القوائم التالية :

File	القائمة الفرعية
Edit	القائمة الفرعية
View	القائمة الفرعية
Project	القائمة الفرعية
Format	القائمة الفرعية
Debug	القائمة الفرعية
Run	القائمة الفرعية
Query	القائمة الفرعية
Diagram	القائمة الفرعية
Tools	القائمة الفرعية
Add-Ins	القائمة الفرعية
Window	القائمة الفرعية
Help	القائمة الفرعية

***الغرض من القوائم :**

بصفة عامة مساعدة المستخدم لهذا البرنامج فى كتابة برامجه، حيث تحتوى هذه القوائم على جميع الأوامر والتعليمات التى تلزم المستخدم للتعامل مع برنامج البيسيك المرئى، حيث تعتبر هذه القوائم مجتمعة هى معدل سطور ومترجم اللغة التى تساعد معد أو مخطط البرامج للوصول إلى الهدف الذى يسعى إليه.

وبالتالى يجب على جميع المستخدمين لهذا البرنامج التعرف على استخدام هذه القوائم، وأيضاً التعرف على التعليمات والوظائف والأوامر التى تحتويها هذه القوائم.

***كيفية التعامل مع القوائم:**

بعد فتح برنامج Microsoft Visual Basic 6.0، واختيار نوع المشروع وليكن Standard EXE يتم ظهور الـ Form الخاص بالمشروع الجديد وهنا يمكن تنفيذ عملية التشغيل أو الوصول إلى هذه القوائم أو تنشيطها من خلال الآتى:

١ - تحريك المؤشر أو السهم الخاص بوحدة الفأرة على اسم القائمة المراد تشغيلها، والضغط عليه بالزر الأيسر للفأرة.

٢ - أو من خلال الضغط على مفتاح ALT مع الحرف الذى يوجد أسفله شرطة، وبالتالى ستفتح لك القائمة الفرعية التى اخترتها ويمكن بالطبع التعامل مع وظائفها أو أوامرها كما سيتضح فيما بعد :

وفيما يلى شرح لكيفية استخدام أوامر القوائم .

٢ – ١ قائمة ملف File

الغرض منها :

التعامل مع الملف أو المشروع (التطبيق)، حيث تحتوى على الأوامر المختلفة التى تساعد المستخدم على التعامل مع الملفات المختلفة التى تنتج عن تنفيذ برنامج VB، ووظيفة هذه القائمة بصفة عامة تتمثل فى فتح المشروع أو التخطيط الجديد لإنشائه والتعامل معه. والشكل التالى يوضح شكل القائمة :

File Edit View Project Format

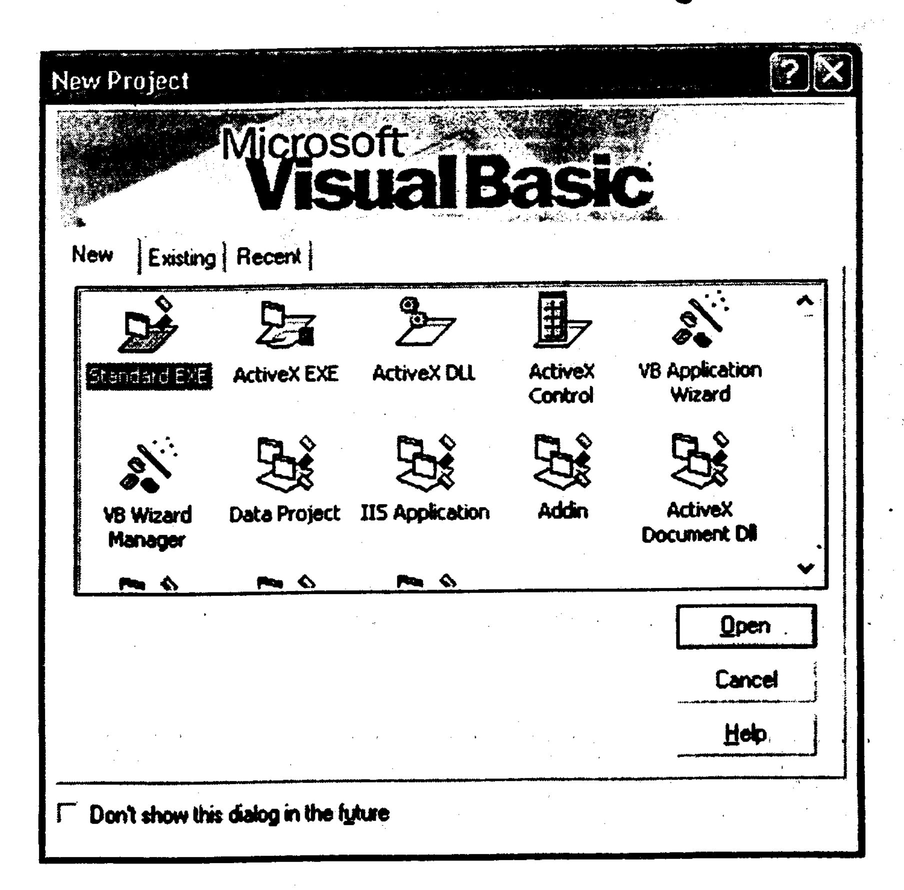
New Project	Ctrl+N
Open Project...	Ctrl+O
Add Project...	
Remove Project	
Save Project	
Save Project As...	
Save Form1	Ctrl+S
Save Form1 As...	
Print...	Ctrl+P
Print Setup...	
Make Project1.exe...	
Exit	Alt+Q

وفيما يلى شرح الوظائف والأوامر الخاصة بهذه القائمة :

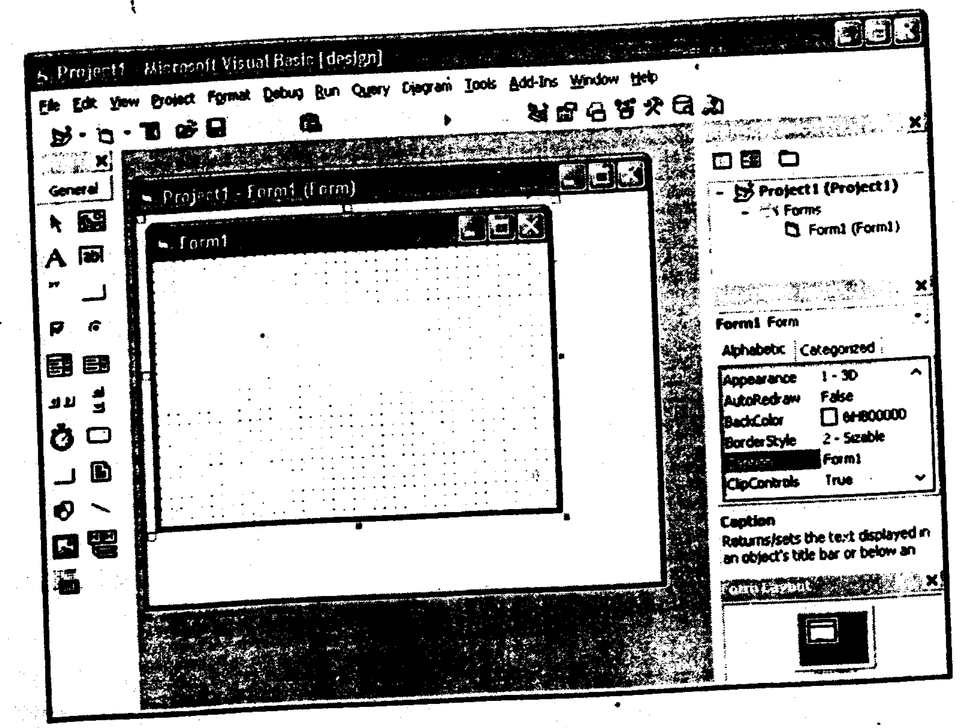
١ - أمر إنشاء مشروع (تخطيط) جديد New Project

الغرض من الأمر :

إنشاء ملف مشروع جديد لإعداد البرنامج الجديد، حيث يتيح هذا الأمر ظهـــور نافذة بأنواع البرامج المختلفة التى يمكنك إنشائها من خلال برنامج Visual Basic والشكل التالى يوضح تلك النافذة:



وبالتالى يمكن اختيـــار Standard EXE وهـــو الاختيـــار الافتراضـــي والرئيسى والذى من خلاله يمكنك الحصول على برنامج تنفيذى، وبتنفيذ هذا الأمر ستظهر الشاشة بالشكل التالى :



ويلاحظ أن النافذة السابقة تتكون من عديد من الأجزاء التى يمكن ايضـــاحها فيما يلى :

١ – نافذة الأوامر Microsoft Visual Basic [Design]
٢ – نافذة الشكل (التصميم) Form
٣ – نافذة صندوق الأدوات Tool Box
٤ – نافذة الخصائص Properties

٥ – نافذة المشروع Project
٦ – نافذة تحكم الألوان Color Palette
٧ – نافذة البرنامج (الكود) Code

✍ ملاحظة :

راجع النوافذ الأخرى التي يمكن التعامل معها من خلال قائمة View.

٢ – أمر فتح مشروع Open Project

الغرض منه :

فتح ملف برنامج (مشروع) سابق إنشاؤه وحفظه من قبل وذلك بغرض التعامل معه من حيث العرض لمحتوياته أو التعديل فيها ...الخ، وبالطبع عند تنفيذ هذا الأمر سيظهر مربع حوارى خاص بعملية الفتح.

كيفية التنفيذ :

فتح قائمة ملف File والتحرك بالمؤشر على الفتح Open Project الضغط عليه من خلال الزر الأيسر للفأرة. وبالتالى سيظهر المربع الحوارى الخاص بهذا الأمر، انظر الشكل التالى:

- المحتويات الخاصة به :

١ – قسم إسم الملف المراد فتحه File name.
٢ – قسم أسماء الملفات القديمة السابق إنشاؤها Existing.
٣ – قسم أسماء الأدلة الموجودة على الوسط التخزيني الحالي.
٤ – قسم عرض أنواع الملفات التى تريد التعامل معها (فتحها حاليا) Files in type.
٥ – قسم وحدات الإدارة الملحقة بالجهاز (الثابتة والليزر والمرنة) Look in.
٦ – أزرار الفتح (Open) أو الإلغاء (Cancel).

* **كيفية التعامل معه :**

لابد من إتباع الآتي :

١- تحديد وحدة الإدارة التى تريد فتح الملف من عليها، من خلال قسم رقم ٥ وحدات الإدارة.
٢- ثم تحديد الدليل الذى تريده أو الذى يحتوى على ملف المشروع المراد فتحة، من خلال قسم ٣ أسماء الأدلة.
٣- ثم تحديد نوعية الملفات المراد فتح إحداها، من خلال قسم رقم ٤ بتحديد النوعية المطلوبة.
٤- ثم اختيار اسم الملف المراد فتحه، من خلال قسم رقم ٢ بتحديده أو تحريك المؤشر عليه، أو فى حالة معرفتك لاسم الملف يمكنك كتابته مباشرة فى قسم اسم الملف، من خلال رقم ١.
٥- ثم اختيار أمر موافق أو النقر على عليه من خلال مؤشر الفارة أو بعد كتابته فى قسم ١ يمكن الضغط على مفتاح الإدخال Enter مباشرة.
٦- وبالتالى سيفتح اسم المشروع المختار وتعرض محتوياته على الشاشة ويمكنك التعامل معه كما تريد .

ملاحظة :

فى حالة الرجوع فى تنفيذ الأمر يمكن تحريك المؤشر على زر Cancel والنقر عليه والعودة كما كنت قبل تنفيذ الأمر

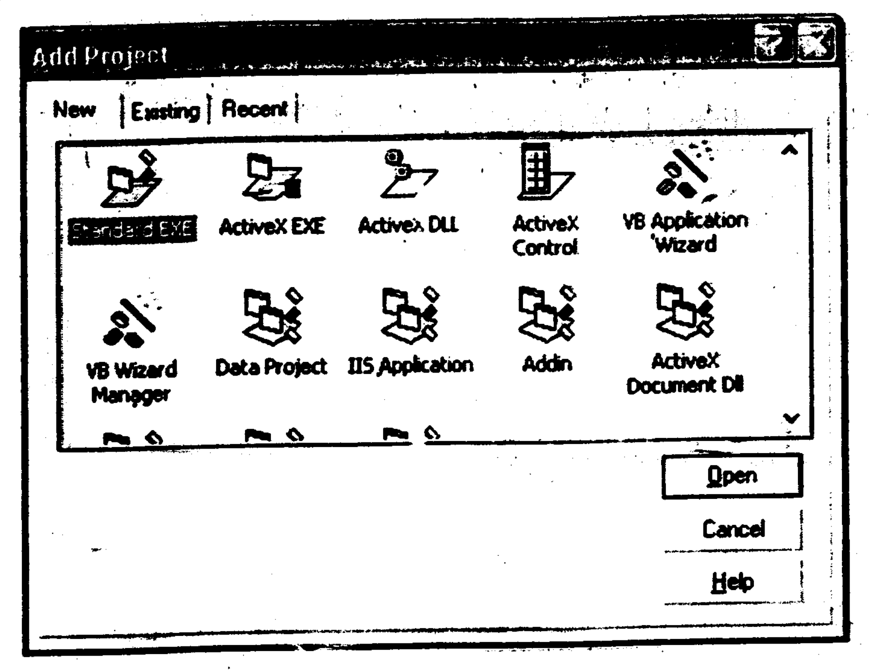
٣-أمر إضافة مشروع جديد Add project :

الغرض منه :

إضافة مشروع Project جديد داخل نافذة المشروع الحالى.

تنفيذ الأمر :

عند اختيار الأمر ... Add Project من قائمة File تظهر النافذة التالية:



فيتم تحديد المشروع Project الجديد ثم الضغط على الزر Open.

٤ – أمر حذف مشروع Remove Project :

الغرض منه :

حذف المشروع النشط حاليا.

تنفيذ الأمر :

عند اختيار الأمر Remove Project من قائمة File تظهر النافذة التالية في حالة عدم حفظ المشروع النشط حاليا ومنها يتم الضغط على Yes للحفظ و No للحذف دون حفظ:

Microsoft Visual Basic

Save changes to the following files?

Project2
Form1

Yes
No
Cancel

٥ – أمر حفظ المشروع الحالي Save Project :

الغرض منه :

حفظ المشروع الجارى التعامل معه أى الملف المفتوح، أى الحفظ على نفس اسم المشروع المفتوح.

ويتم ذلك بغرض الإضافة إليه أى تعديلات أو إضافات قمت بعملها فى الوقت الحالى، وبالتالى يمكن القول أننا نقوم بتخزين سريع.

تنفيذ الأمر :

فتح قائمة ملف File واختيار الأمر من القائمة.

ملاحظة :

وينفذ هذا الأمر لأي نوعية من الملفات المستخدمة فى برنامج Visual Basic.

٦ - أمر حفظ المشروع باسم ... Save Project As :

الغرض منه :

تأمين وحفظ المشاريع والبرامج الفرعية المختلفة الخاصة بالمشروع الحالى (منها ملفات نافذة المشروع ، ملفات البرامج Modules ، ملفات تحكم الأدوات) وأي تعديلات تطرأ عليها أثناء التعامل معها، وذلك فى ملف جديد من نفس النوعية،أي عمل نسخة احتياطية من الملفات المفتوحة بالكامل فى ملف جديد أخر تحت اسم جديد،وبالتالى يجب تنفيذ هذا الأمر بصفة مستمرة أثناء تصميم ملفات نوافذ البرنامج أو البرنامج حتى يكون عندك أكثر من نسخة من هذا الملف.

التنفيذ :

يمكن التنفيذ من خلال الآتي :-

بعد إضافة أى جديد للملف الحالى، يمكن فتح قائمة ملف واختيار هذا الأمر منها؛ وهنا سيظهر مربع حوارى خاص كما يلى :

Save Project As
Save in: VB98
Template
Wizards
File name: Project1.vbp
Save as type: Project Files (*.vbp)
Save
Cancel
Help

عندئذ يتم تحديد وحدة الإدارة المراد الحفظ عليها، ثم تحديد اسم الدليل أيضاً المراد الحفظ عليه، ثم تحديد نوعية الملف، ثم كتابة أو اختيار اسم الملف الجديد المراد الحفظ عليه ثم الضغط على الزر Save.

وهنا يجب تنفيذ هذا الأمر بصفة مستمرة لأن انقطاع التيار الكهربائي سيؤدى إلى فقد التعديلات أو الإضافات التى لم تحفظ فى هذا الملف، حتى نضمن أن يكون هناك ملفين مختلفين لنفس الملف.

٧ – أمر حفظ النموذج Save Form1.frm :

الغرض منه :

تأمين النموذج الحالى وأي تعديلات تطرأ عليه أثناء التعامل مع المشروع الجارى التعامل معه.

وبالتالى يجب تنفيذ هذا الأمر بصفة مستمرة أثناء تصميم البرنامج أو (المشروع) .

تنفيذ الأمر :

أثناء التعامل مع الملف (الإضافة أو التعديل فيه ...) يمكن تنفيذ الأتى :

أ – الضغط على مفتاحى Ctrl + S

ب – فتح قائمة File والتحرك بالمؤشر على أمر ..Save Form والضغط مره واحدة على الأمر.

ولذا يجب تنفيذ هذا الأمر بصفة مستمرة لأن انقطاع التيار الكهربائى سيؤدى إلى فقد التعديلات أو الإضافات التى لم تحفظ فى المشروع الحالى.

٨ - أمر حفظ النموذج باسم ...Save Form1.frm As

الغرض منه :

تأمين وحفظ النموذج الحالى باسم جديد أي عمل نسخة احتياطية من النموذج المفتوح تحت اسم جديد.

التنفيذ :

يمكن التنفيذ من خلال الآتي :

فتح قائمة ملف واختيار هذا الأمر منها، وهنا سيظهر مربع حوارى خاص بذلك لتحديد وحدة الإدارة المراد الحفظ عليها، ثم تحديد اسم الدليل أيضاً المراد الحفظ عليه، ثم كتابة أو اختيار اسم المشروع الجديد المراد الحفظ عليه،كما يلي:

وأيضاً يجب تنفيذ هذا الأمر بصفة مستمرة لأن انقطاع التيار الكهربائي سيؤدى إلى فقد التعديلات أو الإضافات التى لم تحفظ فى المشروع، حتى نضمن أن يكون هناك ملفين مختلفين لنفس المشروع.

٩ - أمر الطباعة Print :

الغرض منه :

طباعة برنامج Code أو Module أو شكل نافذة البرنامج Form على الطابعة.

التنفيذ :

أيضاً من خلال نفس القائمة يمكن اختيار الأمر Print أو يمكن الضغط على مفتاحى Ctrl + P، وعند تنفيذ الأمر سيظهر على الشاشة الشكل التالى :

Print - Project1

Printer: HP LaserJet 6L

Range

Current Module

Current Project

Print What

Form Image

Code

Form As Text

OK

Cancel

Setup...

Print Quality: High

Print to File

Help

ومن خلال هذا المربع يتبع الآتي :

- تحريك المؤشر على المربع المقابل للأمر أو الوظيفة المطلوبة حيث :
  Form Image تشير لطباعة شكل نافذة البرنامج النشطة على الشاشة.
  Code فتشير إلى طباعة الأوامر الخاصة بالبرامج (أوامر لغة البيسيك المرئى).

Form As Text تشير إلى طباعة البيانات النصية والخصائص الخاصة بالأدوات الموجودة على نافذة البرنامج النشطة.

- النقر بالزر الأيسر للفأرة مره واحدة، عندئذ ستظهر علامة صح في المربع.
- وأخير يتم اختيار أمر موافق Ok، لتتم الطباعة.

١٠ - أمر إعداد الطباعة Print Setup

الغرض منه :

إمكانية تحديد مواصفات الإعداد الخاصة بالطابعة الملحقة بالجهاز، حتى تتوافق الطابعة مع برنامج VB.

التنفيذ :

أيضاً من خلال نفس القائمة يمكن اختيار الأمر Print Setup، عندئذ سيظهر المربع الحوارى التالى :

Print Setup
Printer
Name: HP LaserJet 6L
Properties...
Status: Ready
Type: HP LaserJet 6L
Where: LPT1:
Comment:
Paper
Size: A4
Source: Automatically Select
Orientation
Portrait
Landscape
Network...
OK
Cancel

* من خلاله يمكن تنفيذ الآتي :

- تحديد الطابعة الافتراضية الملحقة بالجهاز الحالى باختيار اسمها كما يظهر من الشكل السابق.
- يمكن الضغط على زر Properties للطابعة لتحديد الخصائص الخاصة بها.
- أيضاً من خلال هذا المربع الحوارى يمكن تحديد اتجاه الطباعة (عمودي أو أفقي) بتنشيط الدائرة المقابلة للأمر.
- أيضاً من خلال هذا المربع الحوارى يمكن تحديد حجم الورق ومصدره.
- ثم فى النهاية يتم اختيار أمر موافق OK لقبول الإعداد الجديد للطابعة.

١١- أمر إنشاء ملف برنامج تنفيذي Make Project .exe :

الغرض منه :

إنشاء ملف برنامج تنفيذي مكتوب بلغة الآلة، وذو نوعية EXE للمشروع النشط.

التنفيذ :

يتم تنفيذ هذا الأمر عند الانتهاء من تصميم المشروع بالكامل، وذلك للحصول على ملف واحد يحتوى على كل ملفات وبرامج المشروع بصيغة exe التنفيذية.

وبالطبع الاستفادة من مميزات البرامج التنفيذية والتي من أهمها : -

السرعة فى التنفيذ، الأمان والسرية للبرنامج، العمل على أجهزة متعددة دون الحاجة لمصدر البرنامج.

وللتنفيذ يجب عدم وجود أي أخطاء في برامج وملفات المشروع الملحقة به، أي يجب تنفيذ البرنامج أولاً من خلال الترجمة التفسيرية (من خلال قائمة Run) واختيار أمر Start منها، وبالتالي التأكد من خلو

البرنامج من الأخطاء، عندئذ يمكن في حالة الانتهاء من تصميم ومراجعة البرنامج تنفيذ هذا الأمر، وهنا ستحصل على ملف برنامج بنوعية EXE؛ ويمكن أخذ هـذا البرنامج وإعطاؤه لصاحبه حيث يقوم بتنفيذه بدون الحاجة إلى برنـامج البيسـيك المرئى الأصلي، وفيما يلى شكل المربع الحوارى الخاص به :

من خلال هذا المربع يمكن اختيار اسم وحدة الإدارة واسم الدليل واسم الملـف المراد، ثم اختيار أمر موافق Ok للبدء فى تنفيذ هذا الأمر.

١٢- صندوق عرض أسماء الملفات المستخدمة مؤخراً:

الغرض منه :

عرض أسماء ملفات المشاريع السابق إنشاؤها وفتحها للتعامل معها من قبل (أحدث الملفات).

بالإضافة إلى إمكانية اختيار أحد هذه الملفات للتشغيل أي لفتحها بدلاً من تنفيذ أمر الفتح Open من القائمة الفرعية FILE.

التنفيذ :

تحريك المؤشر إلى الملف المراد التعامل معه بهذا الصندوق، ثم النقر عليه مره واحدة، عندئذ سيفتح المشروع المحدد اسمه على الشاشة وبالتالى يمكنك التعامل معه كما تريد .

١٣- أمر الخروج والإنهاء Exit :

الغرض منه :

إنهاء التعامل مع برنامج Visual Basic، وبالتالى إنهاء التعامل مع الملفات أو المشروع الحالى وذلك بغلقها جميعاً بالترتيب قبل إنهاء التعامل مع برنامج VB.

تنفيذ الأمر :

يجب تنفيذ هذا الأمر عند إنهاء التعامل مع برنامج VB ، وذلك لأنه يعتبر الخروج الطبيعي من البرنامج، وبالتالى فإنك بذلك تحافظ على سلامة البرنامج ككل.

ويتم تنفيذ الأمر كما يلى :

أ- من خلال قائمة File واختيار أمر Exit.

ب- من خلال الضغط على مفتاحى Alt + F4 من لوحة المفاتيح.

ج- من خلال علامة X الموجودة بأعلى يمين البرنامج بشريط العنوان.

د- من خلال مربع التحكم الموجود بأعلى يسار شريط العنوان للبرنامج.

حيث يتم إنهاء التعامل مع النوافذ المفتوحة أو الملفات فى المشروع النشط واحدة تلو الأخرى (متتالية) حتى يتم الخروج النهائى من البرنامج.

٢ - ٢ قائمة تحرير Edit :

الغرض منها :

هذه القائمة خاصة بعمليات التحرير والتعديل المختلفة التى يمكن أن تجرى على ملف المشروع (سواء البرامج أو النوافذ) مثل عمليات النسخ والقص واللصق والإلغاء والبحث والإحلال ..الخ ، وفيما يلى شكل لهذه القائمة:

Edit View Project Format Debug
Undo Insert Text Ctrl+Z
Can't Redo
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Paste Link
Remove
Delete Del
Delete Table from Database
Select All Ctrl+A
Select All Columns
Table
Find... Ctrl+F
Find Next F3
Replace... Ctrl+H
Indent Tab
Outdent Shift+Tab
Insert File...
List Properties/Methods Ctrl+J
List Constants Ctrl+Shift+J
Quick Info Ctrl+I
Parameter Info Ctrl+Shift+I

وفيما يلى شرح لأهم هذه الأوامر:

١ - أمر الرجوع فى تنفيذ الأمر السابق Undo :

الغرض منه :

الرجوع فى تنفيذ الأمر السابق، أى يستخدم مع جميع العمليات المختلفة الخاصة بالبرنامج أو نافذة البرامج والسابق تنفيذها، سواء بالنسخ أو النقل أو الإلغاء أو الإضافة، وتريد العودة إلى الحالة السابقة للتنفيذ (الرجوع فى الأمر) .
معنى ذلك أن هذا الأمر يتغير وظيفته حسب حالة الأمر السابقة .

التنفيذ :

- فى حالة تنفيذ أى عملية (إلغاء سطر مثلاً).
- فتح قائمة التعديل Edit ، واختيار أمر Undo
- عندئذ سيتم الرجوع فى تنفيذ الأمر السابق.

✍ ملاحظة :

يتغير شكل أو وظيفة الأمر فى القائمة حسب حالة التنفيذ التى قمت بها من قبل (أخر عملية)، ويمكن تنفيذ الأمر أيضاً باستخدام مفتاحى Ctrl + Z بدلاً من فتح القائمة.

٢ - أمر تكرار الأمر السابق Redo :

الغرض منه :

تكرار تنفيذ أمر معين (عكس الأمر السابق).

التنفيذ :

- فتح قائمة Edit بعد تنفيذ أمر معين.
- النقر على الأمر بزر الفارة الأيسر.

٣ - أمر القص Cut :

الغرض منه :

قص أو قطع جزء معين من ملف البرنامج (سواء كلمة أو سطر أو مقطع أو أكثر) أو جزء من نافذة البرنامج (أداة) أو الكل بالنسبة لهما ، ووضعها فى حافظة البرنامج (بالذاكرة الرئيسية) .

وذلك لحين نقلها لمكان أخر سواء فى نفس الملف أو لملف أخر أو إلغاؤها من مكانها الحالى .

التنفيذ :

لابد من إتباع الآتي :

١ - بالطبع يجب تحديد الجزء المراد قصه من الملف أو النافذة الحالية، وذلـك بنفس أسلوب التعامل مع البرامج المدرجة فى بيئة النوافـذ، علـى سـبيل المثال لا الحصر (تحريك مؤشر الفأرة إلى بداية المقطع المـراد قصـة (قطعة)، والنقر بالزر الأيسر للفأرة مع السحب لليسار إلى نهاية الجزء أو المقطع الذى تريده، ثم ترك أو تحرير زر الفأرة الأيسر، عندئذ تم التحديد).

٢ - تنفيذ أمر القص من خلال هذه القائمة بتحريك المؤشر علـى الأمـر بهـا والنقر عليه.

٣ - وبالتالي تم تنفيذ الأمر وتم إلغاء أو قص الجزء المحدد .

✍ **ملاحظة :**

يمكن تنفيذ الأمر بواسطة مفتاحى Ctrl + X بدلا من فتح القائمة.

يمكن استخدام أمر اللصق Paste مع هذا الأمر.

٤ - أمر النسخ Copy :

الغرض منه :

نسخ جزء معين من ملف البرنامج (سواء كلمة أو سطر أو مقطع أو أكثر) أو جزء من نافذة البرنامج أو الكل بالنسبة لهما ، ووضعها فى حافظة البرنامج (بالذاكرة الرئيسية) . وذلك لحين النسخ لمكان أخر سواء فى نفس الملف أو لملف أخر .

التنفيذ :

لابد من إتباع الآتي :

١ - بالطبع يجب تحديد الجزء المراد نسخه من الملف أو النافذة الحالية ، وذلك بنفس أسلوب التعامل مع البرامج المدرجة فى بيئة النوافذ ، على سبيل المثال لا الحصر (تحريك مؤشر الفأرة إلى بداية المقطع المراد نسخة، والنقر بالزر الأيسر للفأرة مع السحب لليسار إلى نهاية الجزء أو المقطع الذى تريده، ثم ترك أو تحرير زر الفأرة الأيسر، عندئذ تم التحديد).

٢ - تنفيذ أمر النسخ Copy من خلال هذه القائمة بتحريك المؤشر على الأمر بها والنقر عليه.

٣ - الانتقال إلى الجزء أو المكان الجديد المراد النسخ إليه، بتحريك مؤشر الفأرة إليه والتثبيت له فيه بالضغط مره واحدة .

٤ - عندئذ يمكن تنفيذ أمر أخر للصق لمحتويات الحافظة Paste، وفيما يلى شرحه على حدة.

٥ - وبالتالي تم تنفيذ الأمر وتم النسخ فى الحافظة الخاصة بالبرنامج لحين تكملة الأمر.

✍ ملاحظة :

يمكن تنفيذ الأمر بواسطة مفتاحى Ctrl + C بدلا من فتح القائمة.

٥ - أمر اللصق Paste :

الغرض منه :

لصق جميع محتويات حافظة البرنامج السابق حفظها بواسطة أمر القطع Cut أو أمر النسخ Copy.

التنفيذ :

لابد من إتباع الآتي :
١ - بالطبع تم تحديد الجزء المراد نسخه من الملف أو النافذة الحالية بتنفيذ أمر النسخ أو أمر القطع له، وأصبح هذا الجزء موجود فى الذاكرة الرئيسية.
٢ - الانتقال إلى الجزء أو المكان الجديد المراد النسخ إليه أو اللصق فيه، بتحريك مؤشر الفأرة إليه والتثبيت له فيه بالضغط مره واحدة.
٣ - فتح القائمة Edit والتحرك بالمؤشر لأمر Paste وستجد الأمر نشط (أى يوجد بالحافظة محتويات).
٤ - عندئذ يمكن تنفيذ أمر اللصق Paste لمحتويات الحافظة.
٥ - وبالتالي تم تنفيذ الأمر وتم النسخ على سبيل المثال لكل ما فى الحافظة الخاصة بالبرنامج.

✍ ملاحظة :

يمكن تنفيذ الأمر بواسطة مفتاحى Ctrl + V بدلا من فتح القائمة.
هذا المر قد يظهر بلون خافت (أى لا يمكن تنفيذه) وذلك لعدم وجود محتويات بالحافظة الخاصة بالبرنامج ، وبالتالى لا ينفذ هذا الأمر إلا فى حالة تنفيذ أمر مثل Copy أو امر Cut .

٦- أمر اللصق للتطبيقات الأخرى Paste Link

الغرض منه :
لصق أى محتويات خاصة بالبرامج الأخرى والمراد إدراجها داخل برنامج VB.
معنى ذلك عمل ربط Link بين أى محتويات (مثل الصور) وبين برنامج نافذة الكود الحالى .
وبالتالى يمكن إجراء تعديلات على على المشروع أو على نافذة البرنامج بالإضافة إليها .

٧ - أمر الإلغاء Delete :

الغرض منه :

إلغاء أى محتوى أو جزء من أوامر البرنامج أو من على نافذة البرنامج النشطة بعد تحديده.

التنفيذ :
تنشيط أو تحديد المكان الذى تريد إلغاؤه ، ثم تنفيذ الأمر من القائمة.

✍ **ملاحظة :**

يمكن تنفيذ الأمر بواسطة مفتاح Del بدلا من فتح القائمة.

٨- أمر تحديد الكل Select All :

الغرض منه :
تحديد جميع محتويات النموذج أو نافذة الكود.

التنفيذ : ـ

يمكن التنفيذ من خلال قائمة Edit أو من خلال الضغط على مفتاحى Ctrl+A من لوحة المفاتيح.

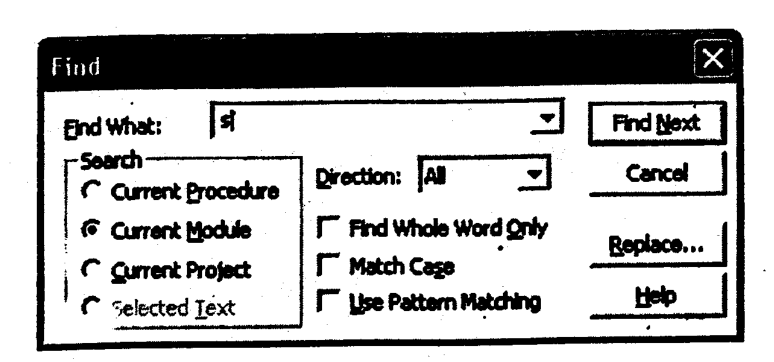
٩ – أمر البحث Find

الغرض منه :

البحث فى نافذة الكود الحالية (Code) عن كلمة معينة أو نص معـــين، وذلك لتنفيذ هدف معين وليكن التعديل فيه.

التنفيذ :

من خلال قائمة التعديل أو التحريــر Edit ، يــتم تحريــك المؤشــر علــى الأمر Find والنقر عليه، عندئذ سيظهر المربع الحوارى أو النافذة التالية :



- وبالتالى يجب تحريك المؤشر فى الصندوق المجاور لكلمتى Find What.
- ثم كتابة الكلمة أو النص المراد إيجاده أو البحث عليه.
- ويمكن تحديد مكان البحث من خلال تحريك المؤشر إلى إحدى الدوائر الملاصقة للأوامر الموجودة أسفل أمر Search .

- ويتم البحث فى الأجراء الحالى Procedure أو البرنامج الحـــالى Module أو المشروع الحالى Project .
- هذا بالإضافة إلى تحديد اتجاه البحث (المكان) Direction ، هل فى كل All مكان البحث أو UP أو Down .
- فى حالة نجاح البحث أول مره يمكن تكرار البحث التالى من خلال الضغط على زر Find Next ، وهكذا .

✍ ملاحظة :

يمكن تنفيذ الأمر بواسطة مفتاحى F + Ctrl بدلا من فتح القائمة وذلك اثناء فتح نافذة الكود Code.

١١- أمر البحث عن التالى Fine Next :

الغرض منه :

استكمال البحث بنفس الشروط التى وضعت فى الأمر السابق.

التنفيذ :

من خلال القائمة أو من خلال الضغط على مفتاح F3 من لوحة المفاتيح.

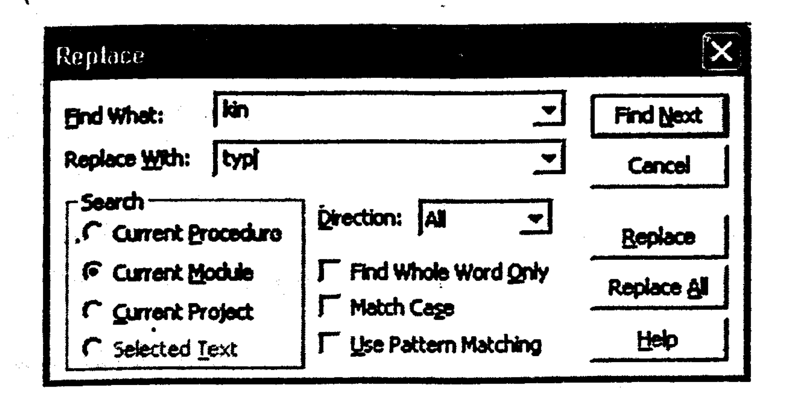
١٢- أمر الاستبدال والإحلال Replace :

الغرض منه :

إجراء عملية إحلال نص معين مكان نص أخر فى نافذة الكود (البرنامج).
حيث يتم البحث أولا فى نافذة الكود الحالية (Code) عن الكلمة أو الـــنص ، وذلك لتنفيذ هدف معين وليكن التعديل فيه أو الإحلال لقيمة جديدة مكان القديمة .

التنفيذ :

- من خلال القائمة الحالية Edit ، يمكن اختيـــار الأمـــر Replace ، وبالتـــالى ستظهر النافذة التالية :



- عندئذ يمكن كتابة الكلمة أو النص المراد البحث عنها أولاً فى المكان المحـــدد للأمر Find What.
- ثم كتابة الكلمة أو النص المراد الإحـــلال لـــه فـــى المكـــان المحـــدد للأمـــر Replace with.
- وأيضاً يمكن تحديد مكان البحث من خلال تحريك المؤشر إلى إحـــدى الـــدوائر الملاصقة للأوامر الموجودة أسفل أمر Search.
- ويتم البحث فى الأجراء الحالى Procedure أو البرنـــامج الحـــالى Module أو المشروع الحالى Project.
- هذا بالإضافة إلى تحديد اتجاه البحث (المكان) Direction، هـــل فـــى كـــل مكان البحث All أو UP أو Down.
- فى حالة نجاح البحث أول مره يمكن تكرار البحث التالى من خلال الضغط على زر Find Next لعمل الإحلال الثانى وهكذا.

✍ ملاحظة :

يمكن تنفيذ الأمر بواسطة مفتاحى H + Ctrl بدلا من فتح القائمة .

١٣- أمر إضافة مسافة بادئة Indent :

الغرض منه :

إضافة مسافة بادئة لسطر معين داخل نافذة الكود.

التنفيذ:

الوقوف على السطر المطلوب إضافة مسافة بادئة له ثم تنفيذ الأمر من القائمة، أو من خلال الضغط على مفتاح Tab من لوحة المفاتيح.

١٤- أمر حذف مسافة بادئة Outdent :

الغرض منه :

حذف مسافة بادئة لسطر معين داخل نافذة الكود.

التنفيذ:

الوقوف على السطر المطلوب حذف مسافة بادئة منه ثم تنفيذ الأمر من القائمة، أم من خلال الضغط على مفتاح Shift مع مفتاح Tab من لوحة المفاتيح.

١٥- أمر إضافة ملف ...Insert File :

الغرض منه :

إضافة ملف داخل نافذة كود البرنامج، ويفيد هذا الأمر فى إمكانية إضافة ملف نصى TXT او Bas أو cls سابق الكتابة مباشرة إلى الكود الحالى.

التنفيذ:

الوقوف على الموضع المطلوب إضافة الملف به ثم تنفيذ الأمر من القائمة، بتظهر النافذة التالية فيتم تحديد الأمر ثم الضغط على الزر Open للتنفيذ.

١٦- أمر عرض قائمة الخصائص والإجراءات List Properties/Methods:

الغرض منه :

عرض قائمة على نافذة الكود Code بجميع الخصائص والإجراءات.

التنفيذ:

الوقوف على الموضع المراد عرض القائمة به من نافذة الكود ثم تنفيذ الأمر من قائمة Edit أو من خلال الضغط على مفتاحى Ctrl + j من لوحة المفاتيح ليظهر شكل قريب من الشكل التالى:

```
Abs
ActiveControl
AlignConstants
AlignmentConstants
App
AppActivate
Appearance
```

ويمكن النقر المزدوج على أى من تلك الخصائص والإجراءات لإضافتها داخل صفحة الكود الحالية.

١٧- أمر معلومات سريعة Quick info :

الغرض منه :

عرض رسالة بها معلومات سريعة عن الأمر الحالى وكيفية كتابته بشكل سليم.

التنفيذ:

الوقوف على الأمر المراد عرض معلومات عن القاعدة العامة له، ثم تنفيذ الأمر، ليظهر شكل مشابه للشكل التالى:

```
msgbox(
 MsgBox(Prompt, [Buttons As VbMsgBoxStyle = vbOKOnly], [Title], [HelpFile], [Context])
 As VbMsgBoxResult
End Sub
```

وهو خاص بعرض القاعدة العامة لأمر msgbox( ).

ولإخفاء هذه المعلومات يمكن النقر بعيداً عن سطر الأمر.

٢ - ٣ قائمة عرض View :

الغرض منها :

هذه القائمة خاصة بعرض وإخفاء نوافذ وأشرطة الأدوات للبرنامج وبعمليــات تحسين الأداء للمستخدم ومساعدته على أداء وظائف مرتبطة بهذه القائمة والمشاهدة الخاصة بالبرنامج ومساعدة المستخدم فى إعداد أوامر البرنامج مثل أمــر Code وإظهار نافذة Form للتعامل معها من خلال أمر Form ...الخ.

وبالتالى يمكن القول أنها قائمة مرتبطة بإعداد برنامج المستخدم من خلال VB والشكل التالى يوضح شكل تلك القائمة:

View Project Format Debug Run

Code
Object Shift+F7

Definition Shift+F2
Last Position Ctrl+Shift+F2

Object Browser F2

Immediate Window Ctrl+G
Locals Window
Watch Window
Call Stack... Ctrl+L

Project Explorer Ctrl+R
Properties Window F4
Form Layout Window
Property Pages Shift+F4

Table ▸
Zoom ▸
Show Panes ▸

Toolbox
Data View Window
Color Palette

Toolbars ▸

Visual Component Manager

وفيما يلى شرح لأهم الأوامر الخاصة بهذه القائمة :

١ - أمر نافذة الكود Code

الغرض منه :

إظهار نافذة الكود الخاصة بكتابة الأوامر المرتبطة بتشغيل نافذة البرنـامج Form أو بالأداة Tool أو بنموذج البرنامج Module .، وبالتالى يمكـن القـول بأنها الوعاء الذى يتم فيه كتابة البرامج الرئيسية والفرعية.

التنفيذ :

أولاً : التنفيذ بالنسبة لنافذة البرنامج Form

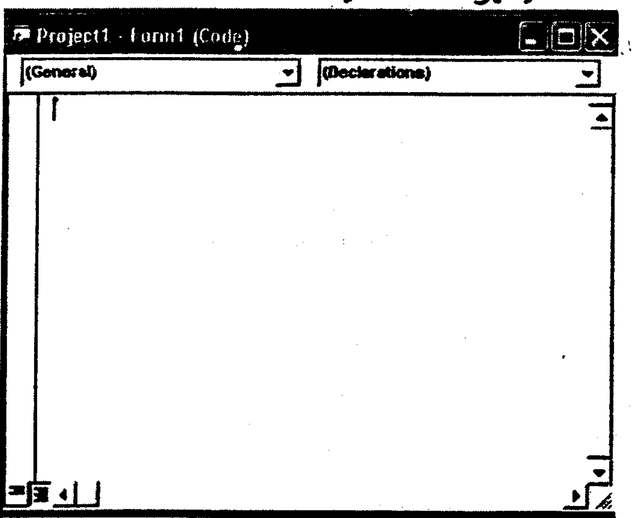
أ - التشغيل من خلال قائمة عرض View

لتنفيذ ذلك يتبع الآتي :

١ - فتح نافذة برنامج جديدة .

٢ - فتح قائمة المشاهدة View ، ثم اختيار أمر Code .

٣ - عندئذ سيظهر النافذة التالية :



٤- بالتالى يمكن كتابة التعليمات الخاصة بتشغيل هذه النافذة كما تريد، كما سنعرف فيما بعد .

ب - التشغيل من خلال النقر المزدوج على زر الفأرة الأيسر :

لتنفيذ ذلك يتبع الآتي :

١ - أيضاً تنشيط نافذة البرنامج المراد كتابة البرنامج لها .
٢ - يمكن النقر المزدوج مباشرة، وبالتالى ستظهر نافذة الكود .
٣ - بالتالى يمكن كتابة التعليمات الخاصة بتشغيل هذه النافذة كما تريد، كما سنعرف فيما بعد .

ج - التشغيل من خلال مفتاح View Code بنافذة المشروع Project

لتنفيذ ذلك يتبع الآتي :

١ - حيث يتم تنشيط نافذة البرنامج أو الأداة أولا.
٢ - التحرك بمؤشر الفأرة إلى نافذة الـ Project والنقر بالزر الأيسر للفأرة على مفتاح أو زر View Code الموجود بها.
٣ - عندئذ ستظهر نافذة الكود الخاصة بالنافذة أو بالأداة النشطة.
٤ - وبالتالى يمكنك التعامل معها بكتابة الأوامر الخاصة بها، وهكذا بالنسبة لأى نوافذ برنامج أو أدوات .

ثانياً : التنفيذ بالنسبة للأدوات Tools

أيضاً يمكن فتح نافذة برنامج جديدة لوضع أدوات برنامج البيسيك المرئى Visual Basic عليها ، ثم تحديد الخصائص الخاصة بتشغيل هذه الأدوات، ثم كتابة التعليمات أو الأوامر اللازمة لتشغيل هذه الأدوات فى نافذة الكود الخاصة بكل أداة.

ولتنفيذ ذلك يتبع الآتي :

أيضاً يمكن التنفيذ من خلال قائمة عرض أو النقر المزدوج أو من خلال نافذة المشروع وبالتالى يمكن التنفيذ مع أداة معينة ومن خلال النقر المزدوج كما يلى :

١ – تنشيط نافذة البرنامج المراد التعامل معها.

٢ – وضع الأداة المطلوب إعداد نافذة الكود لها على نافذة البرنامج Form النشطة.

٣ – وبالتالى يمكن النقر المزدوج مباشرة على شكل أو مكان الأداة التى تحددها أي النشطة ، وبالتالى ستظهر نافذة الكود الخاصة بها.

٤ – عندئذ يمكن كتابة التعليمات الخاصة بتشغيل هذه الأداة، كما سنعرف فيما بعد.

✍ ملاحظة :

يمكن إلغاء أو غلق نافذة الكود من خلال سطر العنوان واختيار أمر إغلاق أو النقر على علامة X بنفس أسلوب التعامل مع البرامج المدرجة فى بيئة النوافذ.

<u>ثالثاً</u> : من خلال لوحة المفاتيح

يمكن تشغيلها أيضاً من خلال الضغط على مفتاح F7 .
وفيما يلى شكل نافذة الكود مع الأدوات :

Project1 - Form1 (Code)

Command1 | Click

```
Private Sub Command1_Click()

End Sub

Private Sub Form_Load()

End Sub
```

٢ - أمر نافذة البرنامج Object

الغرض منه :

إظهار نافذة البرنامج Form النشطة الموجودة داخل نافذة المشروع الحالي.
للبدء فى تصميم القوائم وشاشات الإدخال والإخراج ...الخ .

التنفيذ :

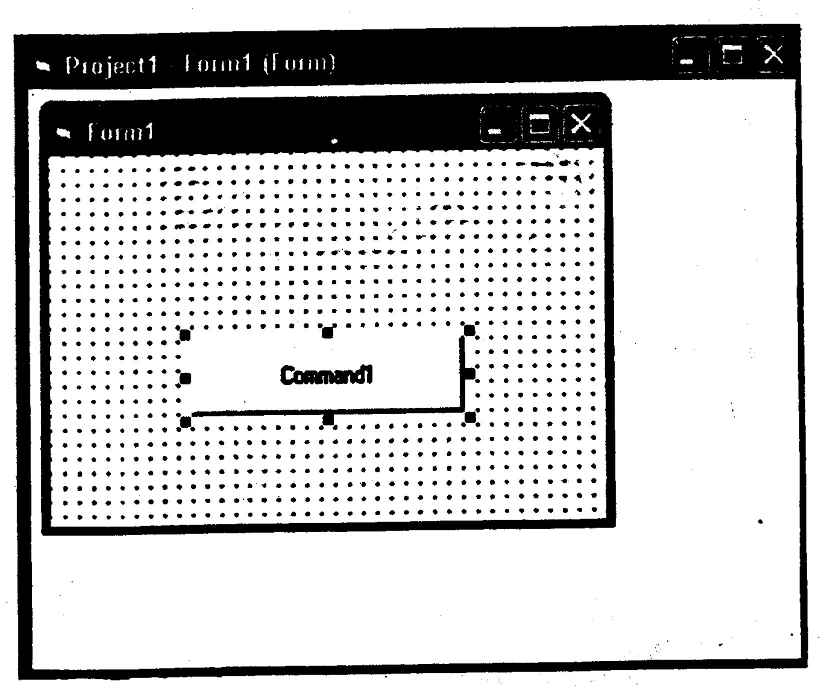
فى حالة عدم تنشيط أي نافذة برنامج.
يمكن تحريك مؤشر الفأرة على اسم أي نافذة موجود فى نافذة المشروع.
ثم يتم التشغيل من خلال إحدى الطرق التالية :

أ – فتح قائمة View واختيار أمر Object منها، عندئــذ ســتظهر نافــذة البرنامج المحددة.

ب – النقر على مفتــاح أو زر View Object الموجــود بنافــذة Project، عندئذ أيضاً ستظهر نافذة البرنامج المحددة (النشطة).

ج – من خلال النقر من لوحة المفاتيح على المفتاحين Shift+F7 معا.

وفيما يلى شكل نافذة البرنامج :



✍ ملاحظة :

قد يكون هناك أكثر من نافذة برنامج فى نافذة المشروع، وبالتالى يمكن تحريك مؤشر الفأرة على إحداها أى تنشيطها، وسوف تحدد باللون الأزرق دليل التنشيط لها.

٣ – أمر مستعرض الكائنات Object Browser :

الغرض منه :

إمكانية عرض أو التعرف على الأوامر المختلفة والمكتبات الخاصـــة بهـــا التى يمكن استخدامها بشكل عام أو مع النماذج السابق إنشـــاؤها، بالإضـــافة إلـــى عرض محتويات كل مشروع على حدة من البرامج والنوافذ وخصـــائص الأدوات المستخدمة فيه.

التنفيذ :

بالطبع فى القائمة View، والتحرك للأمر المراد تنفيذه (الحالى.)، ثم النقـــر عليه، عندئذ سيظهر على الشاشة النافذة الخاصه بهذا الأمر كما يلى :

Object Browser

<All Libraries>

<All Libraries>
Project1
stdole
VB
VBA
VBRUN

Member

Classes
<globals>
AlignConstants
AlignmentConstants
AmbientProperties
App
ApplicationStartCon
AsyncProperty

Members of '<globals>'
Abs
App
AppActivate
Asc
AscB
AscW
Atn

<All Libraries>

- نلاحظ فيها ما يلى :

أ - القائمة المنسدلة الخاصة بالـ Libraries:

حيث يمكن السهم الموجود على شريط التمرير لعرض أسماء المشروعات أو أجزء المراد عرض تفاصيل له، وبالطبع تحديد المشروع المراد التعامل معه.

ب - صندوق أو قائمة Classes:

حيث يحتوى على أسماء النوافذ والبرامج المختلفة المندرجة تحت المشروع النشط أو المفتوح، وأيضاً يمكن التحرك داخل هذا الصندوق واختيار النافذة أو البرنامج المراد التعرف عليها.

ج - صندوق الوسائل والخصائص Methods:

حيث يحتوى هذا الصندوق أو هذه القائمة على الوسائل والخصائص المختلفة الخاصة بنافذة البرنامج أو أى ملف موجود فى القسم السابق (ب).

د - أيضاً يمكن البحث عن أمر معين من خلال الجزء الخاص بالبحث، والذى يظهر بالشكل التالى :

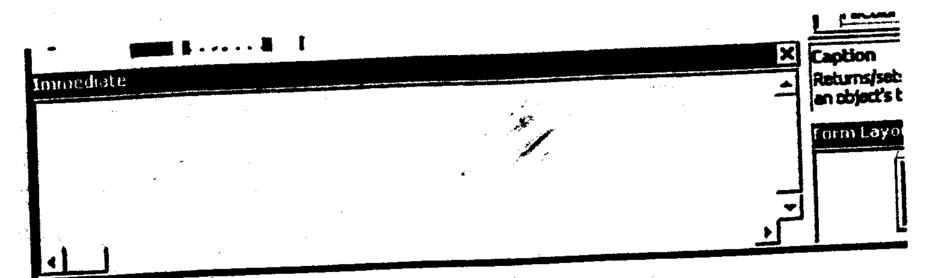
٤ - أمر عرض نافذة Immediate Window

الغرض منه :

إظهار نافذة **Immediate** والتى تظهر بشكل افتراضـــى أســفل نافــذة البرنامج.

التنفيذ :

من خلال النقر عليها من القائمة أو من خلال الضغط على مفتاحي Ctrl+G، عند ذلك يظهر الشكل التالى:



٥ - أمر عرض نافذة Locals Window

الغرض منه :

إظهار نافذة **Locals** والتى تظهر بشكل افتراضى أسفل نافذة البرنامج.

التنفيذ :

من خلال النقر عليها من القائمة، عند ذلك يظهر الشكل التالى:

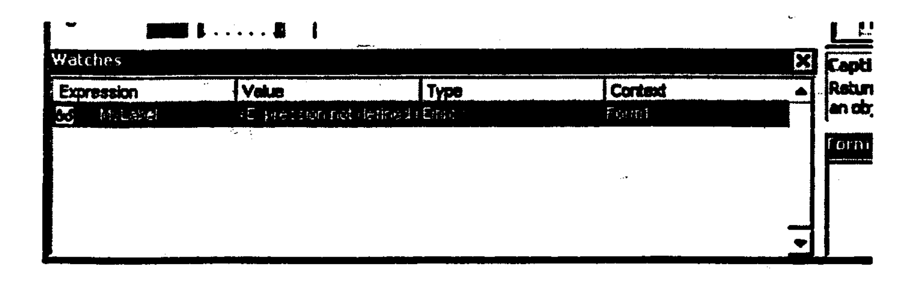
٦ - أمر عرض نافذة Watches Window

الغرض منه :

إظهار نافذة **Watches** والتى تظهر بشكل افتراضى أسفل نافذة البرنامج.

التنفيذ :

من خلال النقر عليها من القائمة، عند ذلك يظهر الشكل التالى:



ويمكن الإضافة لها من خلال قائمة Debug.

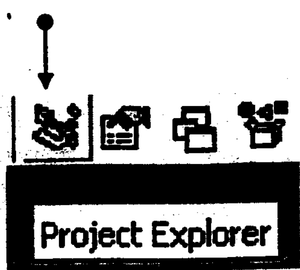
٧ - أمر عرض نافذة مستعرض محتويات البرنامج Project Explorer:

الغرض منه :

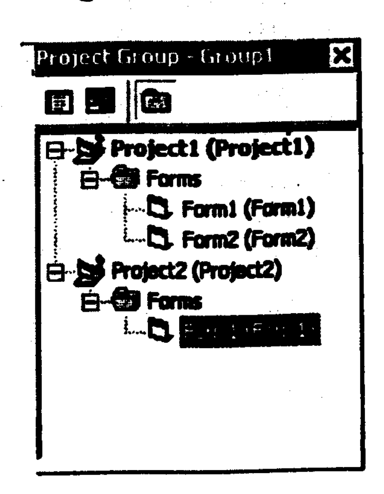
إظهار نافذة تحتوى على قائمة بأسماء المشـــاريع الحاليـــة داخـــل البرنـــامج. بالإضافة لما يحويه كل مشروع من نماذج Forms.

التنفيذ :

يتم الضـــغط علـــى الأمـــر مـــن خـــلال القائمـــة View، أو مـــن خـــلال مفتاحى Ctrl+R من لوحة المفاتيح، أو من خلال شريط الأدوات والضغط علـــى الزر Project Explorer :



وفى جميع الحالات تظهر نافذة الأمر كما يتضح من الشكل التالى:



٨- أمر عرض نافذة خصائص الكائنات Properties Window:

الغرض منه :

إظهار نافذة تحتوى على قائمة بالخصائص المتاحة للعنصر (الكائن) المحدد على النموذج Forms الحالى، مع إمكانية التعديل والتغيير بقيم تلك الخصائص.

التنفيذ :

يتم الضغط على الأمر من خلال القائمة View، أو من خلال مفتاح F4 من لوحة المفاتيح، أو من خلال شريط الأدوات والضغط على الزر Properties Window وعند التنفيذ يظهر الشكل التالى:

Properties - Command1

Command1 CommandButton

Alphabetic | Categorized

(Name)	Command1
Appearance	1 - 3D
BackColor	&H8000000
Cancel	False
Caption	Command1
CausesValidatio	True
Default	False
DisabledPicture	(None)
DownPicture	(None)
DragIcon	(None)
DragMode	0 - Manual
Enabled	True
Font	MS Sans Serif
Height	735
HelpContextID	0
Index	

Caption
Returns/sets the text displayed in

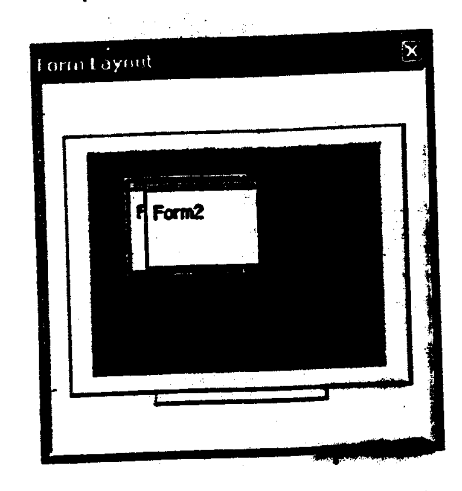
٩- أمر عرض نافذة موضع ظهور النموذج Form Layout Window:

الغرض منه :

إظهار نافذة تحتوي على شكل لشاشة الحاسب الآلى والتى يمكن من خلالها تحديد موضع ظهور النموذج على الشاشة أثناء تنفيذ البرنامج.

التنفيذ :

يتم الضغط على الأمر من خلال القائمة View، أو من خلال شريط الأدوات والضغط على الزر Form Layout Window وعند التنفيذ يظهر الشكل التالى :



ويتم الضغط بزر الفأرة الأيسر على شكل النموذج المراد تحديد موضع له والاستمرار في الضغط حتى يتم الوصول للموضع المطلوب ثم الإفلات ليستقر النموذج فى موضعه الجديد ولن يظهر ناتج تنفيذ الأمر إلا أثناء تشغيل البرنامج إما للتجربة بأمر RUN أو التشغيل النهائى للبرنامج بعد الترجمة إلى EXE.

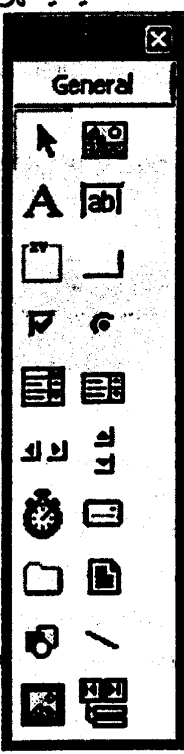
١٠- أمر عرض شريط الأدوات ToolBox:

الغرض منه :

إظهار شريط الأدوات الخاص ببرنامج VB والذى من خلالـه يـتم إضـافة الكائنات والعناصر على النموذج، مثل مربع النص Text Box وصندوق الاختيار Check Box ...الخ.

التنفيذ :

يتم الضغط على الأمر من خلال القائمة View، أو من خلال شـريط الأدوات والضغط على الزر ToolBox عند التنفيذ يظهر الشكل التالى :



ويتم إضافة العناصر منه من خلال النقر المزدوج على رمز العنصر المطلوب، أو من خلال النقر المفرد على الرمز ثم الذهاب للنموذج والنقر عليه مع السحب.

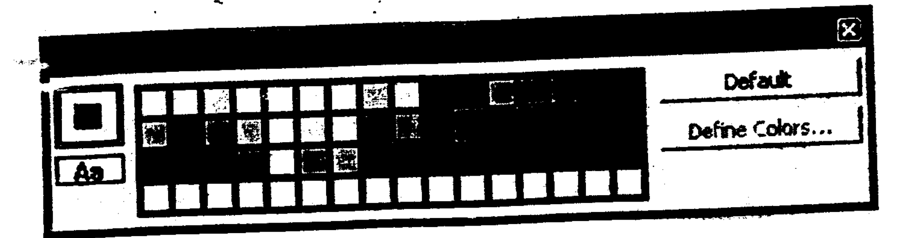
١١- أمر نافذة الألوان Color Palette :

الغرض منه :

إظهار أو عدم إظهار نافذة الألوان أثناء تصميم البرنامج للاستفادة منها في تحديد الألوان لبرنامجك.

التنفيذ :

يتم الضغط على الأمر من خلال القائمة ليظهر الشكل التالي:



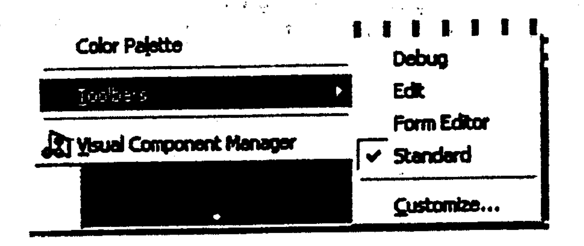
١٢- القائمة الفرعية أشرطة الأدوات ToolBars:

الغرض منها:

وهي خاصة بعرض قائمة فرعية بأسماء أشرطة الأدوات الخاصة بالبرنامج.

التنفيذ :

يتم الوقوف بالفارة على الأمر لتظهر القائمة الفرعية الخاصة به ومنها يمكـــن تحديد القائمة المراد إظهارها أو إخفائها بالضغط عليهـــا مـــرة واحـــدة بالفـــأرة، انظر الشكل التالي:



✍ ملاحظة :

فى حالة وجود علامة صح (✓) بالقائمة السابقة، يعنى ذلـــك أن الأمر نشط وبالتالى صندوق الأدوات **ToolBoxs** ظاهر على الشاشــــة، وبالطبع يمكنك التعامل معه.

✍ ملاحظة : عامة لهذه القائمة (View):

سوف يتم شرح الخطوات التفصـــيلية للتعامـــل مـــع أغلـــب محتوياتها بالتفصيل فيما بعد فى فصـــول مســـتقلة إلـــى حـــد مـــا، وبالتالى يجب عليك معرفة الغرض منها وكيفية فتح أو تنشيط هـــذه النوافذ أو الصناديق، ثم يلى ذلك فى الفصول المستقلة التعرف علـــى كيفية الاستخدام لها والتعامل معها.

٢ - ٤ قائمة المشروع Project :

الغرض منها :

القيام بعملية التحكم فى المشروع من الإدراج أو الإضـــافة داخــل المشـــروع الحالى، حيث يمكن إدراج نافذة برنامج Form أو إدراج نافذة MDI التى تحتوى على أكثــر مـــن نافـــذة برنـــامج Form أو إدراج برنـــامج بيســـيك Module، أو بحذف نموذج أو بفتح نافذة إضافة مكونات جديدة لشريط الأدوات وأيضا نافـــذة خصائص المشروع الحالى ...

وتتضح هذه الأوامر من شكل القائمة التالى:

Project Format Debug Run

Add Form
Add MDI Form
Add Module
Add Class Module
Add User Control
Add Property Page
Add User Document
Add WebClass
Add Data Report
Add DHTML Page
Add Data Environment
Add File... Ctrl+D

Remove Form1

References...
Components... Ctrl+T

Project2 Properties...

التنفيذ :

بالطبع مثل التعامل مع النوافذ الأخرى، وتحتوى هذه القائمة على مجموعة من الأوامر منها:

١ - إضافة نافذة برنامج (نموذج) Add Form :

بالطبع يمكن من خلال البرنامج الحالى إمكانية إدراج نافذة برنامج جديدة تلحق بنافذة المشروع، على أن تحتوى على أدوات جديدة أو برامج فرعية جديدة.
وبالتالى يمكن التنفيذ مباشرة أثناء تصميمك للمشروع الحالى، وهنا ستفتح قائمة الإدراج، واختيار أمر Form، وبالتالى سيظهر على الشاشة نافذة برنامج جديدة، يمكنك التعامل معها (راجع شكل Form).

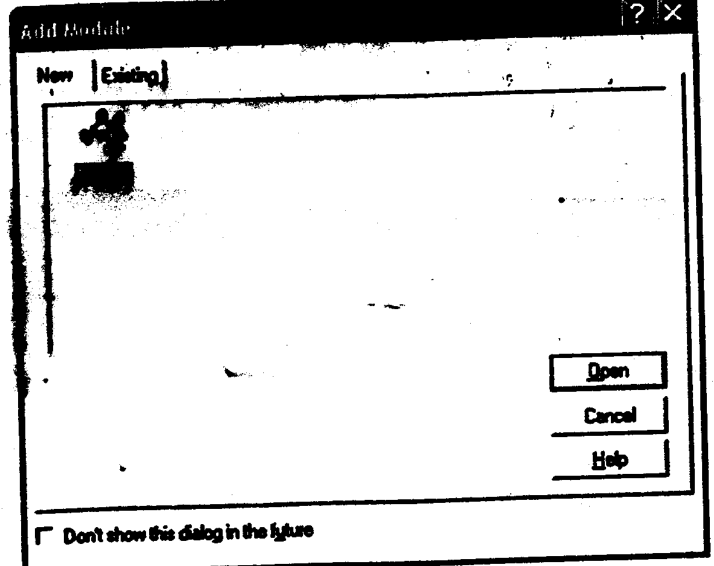
٢ - إضافة نافذة Add MDI Form :

تأخذ هذه النافذة شكل نافذة البرنامج Form ولكن بالطبع النوعية مختلفة، حيث تحتوى هذه على مجموعة من نوافذ Forms داخلها.
مع العلم أن ملف المشروع ذو النوعية Mak لا يستوعب أكثر من نافذة MDI واحدة فقط.
والإدراج لها داخل المشروع الحالى يتم بنفس الأسلوب السابق.

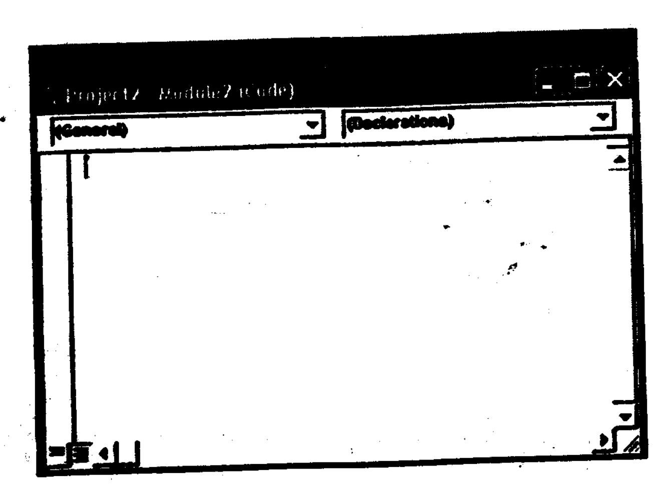
٣ - إضافة نافذة Add Module :

حيث يتم إدراج نافذة ملف برنامج Module ليتم كتابة التعليمات الجديدة أو البرنامج الجديد المراد إدراجه أيضاً داخل نافذة المشروع Project للتعامل معه.
وأيضاً مشابه لهذا الإدراج، إدراج Class Module.

وفيما يلى شكل نافذة الـ Module :

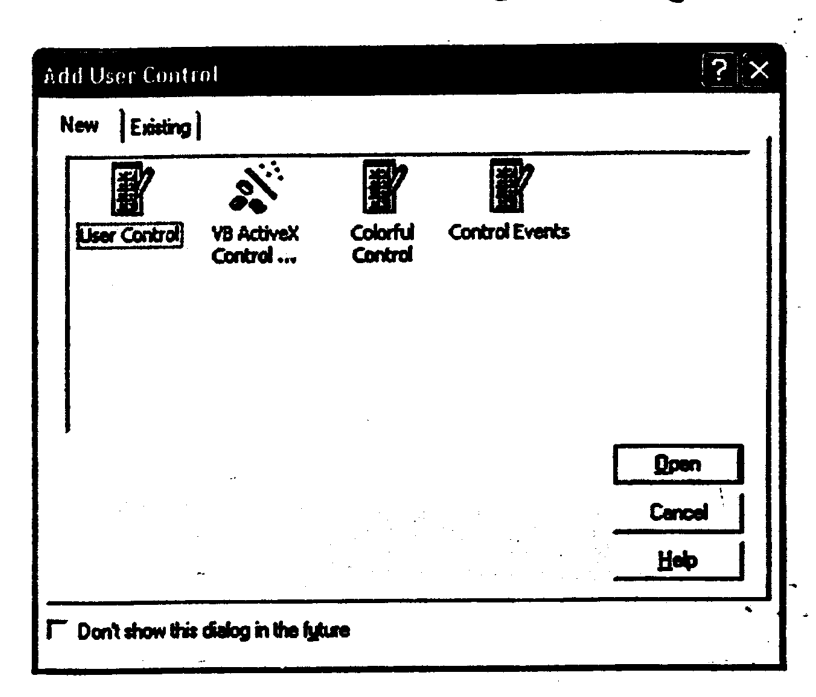


- بتنفيذ أمر Open ستظهر النافذة التالية التي سيتم الكتابة للأوامر بها.



٤ – إضافة أدوات يقوم بتصميمها المستخدم Add User Control :

عند النقر عليها يتم ظهور مربع حوار لاختيار ما يرغب المستخدم فى البدء فى تصميمه من أدوات خاصة به.
ويظهر المربع الحوارى التالى عند التنفيذ:



٥ – إضافة ملف Add File :

يمكن أيضا إدراج ملف معين وليكن ملف نصى إلى المشروع الحالى، وعند تنفيذ الأمر ستظهر النافذة التالية :

Add File

Look in: VB98

Template
Wizards

File name:
Files of type: VB Files (*.frm;*.ctl;*.pag;*.dsr;*.bas;*.cls;*.re:

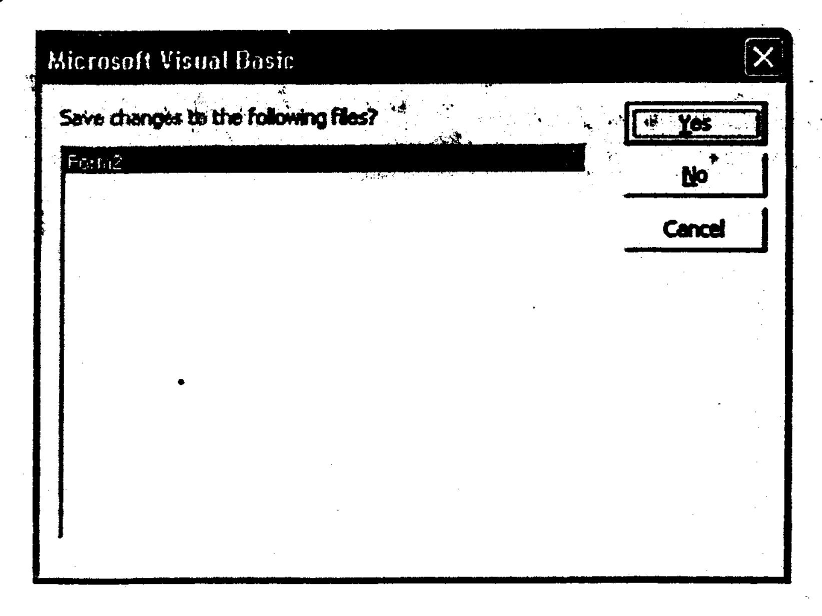
Open
Cancel
Help

Add As Related Document

وبالتالى يمكنك تحديد الوحدة والدليل الموجود عليهم الملف المــراد إدراجــه، ثم تحديد اسم الملف المراد وتنفيذ الأمر.

٦ – حذف نموذج Remove Form :

يتم من خلاله حذف النموذج النشط، وفى حالة عدم حفظ النموذج تظهر النافذة التالية:



وفى حالة الرغبة فى حذف نموذج معين يجب تنشيطه أولا، ثم تنفيذ أمـــر الحذف من القائمة.

٧ - عرض نافذة References:

والتى يتم من خلالها عرض نافذة بالمكتبـــات الخاصـــة بـــالأدوات المتـــوفرة بالبرنامج، ويمكن حذف وإضافة مكتبة جديدة أو حذف أخرى موجودة.
ويتم الحذف والإضافة بوضع علامة صح أمام المكتبة المراد إضافتها أو حذفها من أما ما يراد حذف مكتبته، ويمكن البحث اليدوى على المكتبة مـــن خـــلال زر الأمر Browse؛ والشكل التالى يوضح شكل النافذة:

References - Project1

Available References:

- [x] Visual Basic For Applications
- [x] Visual Basic runtime objects and procedures
- [x] Visual Basic objects and procedures
- [x] OLE Automation
- [ ] Microsoft Data Formatting Object Library
- [ ] Microsoft Data Report Designer v6.0
- [ ] IAS Helper COM Component 1.0 Type Library
- [ ] IAS RADIUS Protocol 1.0 Type Library
- [ ] Active DS Type Library
- [ ] Active Setup Control Library
- [ ] ActiveMovie control type library
- [ ] ActiveX DLL to perform Migration of MS Repository V:
- [ ] adsauth 1.0 Type Library
- [ ] ASFChop 1.0 Type Library

OK | Cancel | Browse... | Priority | Help

OLE Automation

Location: E:\WINDOWS\System32\stdole2.tlb

Language: Standard

٨ - عرض نافذة المكونات Components:

والتى يتم من خلالها عرض نافذة بالأدوات الإضافية المتوفرة للبرنامج، ويمكن حذف وإضافة أى منها من خلال وضع أو إزالة علامة التحديد من أمامها.

ويمكن تنفيذه أيضا بالضغط على مفتاحى Ctrl + T.

والشكل التالى يوضح شكل نافذة الأمر:

٩ - عرض نافذة خصائص المشروع الحالى Project Properties :

ومن خلالها يتم عرض وتعديل النوع والبيانات للمشروع الحالى والخاصة بالبرنامج التنفيذى منه (أى بعد ترجمته)، والاسم الخاص به وأيضا تحديد النموذج الذى سيتم البدء به أثناء التشغيل الفعلى للبرنامج ... الخ.
والشكل التالى يوضح نافذة الأمر:

Project1 - Project Properties

General | Make | Compile | Component | Debugging

Project Type: Standard EXE
Startup Object: Form1
Project Name: Project1
Help File Name:
Project Help Context ID: 0
Project Description:

Unattended Execution
Upgrade ActiveX Controls
Require License Key
Retained In Memory

Threading Model
Thread per Object
Thread Pool 1 threads

OK | Cancel | Help

✍ ملاحظة :

- بالطبع يوجد الكثير من الخيارات المختلفة بهذه النافذة، وبالتالي يجب التعرف على كل منها ، بمعرفة الغرض وطرق الاستخدام، والخيارات الإضافية الفرعية الخاصة بكل منهم، وذلك لزيادة المعرفة الأكثر تخصصاً.

٢ - ٥ قائمة تهيئة Format:

الغرض منها :

التحكم بالعناصر الموجودة على النموذج Form من حيث موضعها وتباعدها فيما بينها لوضعها فى التنسيق والموضع المناسب لوجهة نظر المصمم، مع إمكانية منع التعديل فيها.

والشكل التالى يوضح شكل هذه القائمة:

Format Debug Run Q
Align ▸
Make Same Size ▸
Size to Grid
Horizontal Spacing ▸
Vertical Spacing ▸
Center in Form ▸
Order ▸
Lock Controls

- وفيما يلى شرح لأوامرها:

١- القائمة الفرعية محاذاة Align :

الغرض منها:

التحكم فى موضوع العناصر المتعددة على نافذة البرنامج Form، وهو مفيد لترتيب عناصر النموذج بشكل منتظم.

وتحتوى تلك القائمة خاصة بمحاذاة العناصر بالتوسيط أو يميناً ويساراً المحاذاة أعلى وأسفل وتوسيط رأسى وأخيراً المحاذاة لخطوط شبكة النموذج.

التنفيذ:

يتم تحديد أكثر من عنصر من العناصر الموجودة على النموذج Form بأحد الطرق مثل الضغط على زر Ctrl + زر الفأرة الأيسر على العناصر المطلوب تحديدها، ثم فتح القائمة لتجدها بالشكل التالى، فيتم اختيار الأمر المطلوب منها:

Format Debug Run Query Diagram To
Align
Make Same Size
Size to Grid
Horizontal Spacing
Vertical Spacing
Center in Form
Lefts
Centers
Rights
Tops
Middles
Bottoms
to Grid

٢- القائمة الفرعية ضبط بنفس الحجم Make Same Size :

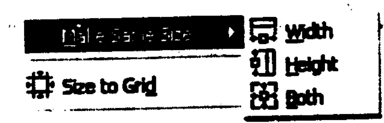
الغرض منها:

ضبط عناصر النموذج المحددة حتى تكون بحجم واحد، أفقى أو رأسى أو الاثنين معاً.

التنفيذ:

يتم تحديد أكثر من عنصر من العناصر الموجودة على النموذج Form بأحد الطرق مثل الضغط على زر Ctrl + زر الفأرة الأيسر على العناصر

المطلوب تحديدها، ثم فتح القائمة لتجدها بالشكل التالى، فيـتم اختيــار الأمــر المطلوب منها:



٣-ضبط العناصر تبعاً لخطوط الشبكة Size to Grid :

الغرض منها:

ضبط حجم العنصر لتكون أطرافه مضبوطة على نقاط خطـوط الشـبكة الخاصة بالنموذج، ويفيد هذا فى إمكانية تنسيق العناصر فى ترتيب متسـاوى أفقى أو راسى.

التنفيذ:

يمكن التنفيذ على عنصر واحد أو على عدة عناصر محددة بطرق التحديـد السابقة.

ملاحظة :

هذا الأمر لا تظهر فائدة إلا فى حالة أن العناصر لا تلتصق بخطوط الشبكة بشكل افتراضى، حيث أن الوضع الافتراضى هو التصاق العناصـر بخطـوط الشبكة للنموذج، ويمكن إلغاء هذا الالتصاق الافتراضى مـن خـلال الـذهاب لقائمة Tools واختيار الأمـر Option والـذهاب للتبويـب General ثـم إلغاء التحديد من الأمر " Align Controls to Grid "، والشكل التالى يوضح الأمر حيث يظهر فى أسفل يسار النافذة :

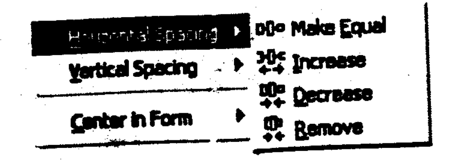
٤- القائمة الفرعية الخاصة بالمسافات الأفقية Horizontal Spacing :

الغرض منها:

ضبط المسافات الأفقية بين عناصر النموذج وبعضها البعض، من حيث جعلها متساوية أو زيادة المسافة أو إنقاصها أو حذفها.

التنفيذ:

يمكن التنفيذ على عنصرين أو عدة عناصر محددة بطرق التحديد السابقة، حيث لا تنشط القائمة إلا بعد تحديد أكثر من عنصر، ويتم اختيار الأمر المطلوب من القائمة تبعاً لرغبة المصمم للبرنامج، والشكل التالى يوضح شكل القائمة الفرعية:



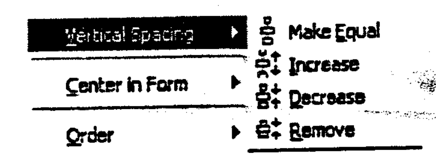
٥- القائمة الفرعية الخاصة بالمسافات الراسية Vertical Spacing :

الغرض منها:

ضبط المسافات الرأسية بين عناصر النموذج وبعضها البعض، من حيث جعلها متساوية أو زيادة المسافة أو إنقاصها أو حذفها.

التنفيذ:

يمكن التنفيذ على عنصرين أو عدة عناصر محددة بطرق التحديد السابقة، حيث لا تنشط القائمة إلا بعد تحديد أكثر من عنصر، ويتم اختيار الأمر المطلوب من القائمة تبعاً لرغبة المصمم للبرنامج، والشكل التالى يوضح شكل القائمة الفرعية:



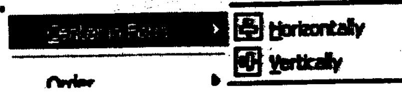
٦- القائمة الفرعية توسيط بالنموذج Center in Form :

الغرض منها:

ضبط العنصر الحالى أو العناصر المحددة لتكون فى وسط النموذج، وذلك من خلال الضبط للعنصر إما أفقيا أو رأسياً.

التنفيذ:

يمكن التنفيذ على عنصر أو عنصرين أو عدة عناصر محددة بطرق التحديد السابقة، ثم تنفيذ الأمر من القائمة ليتم تحريك العنصر لوسط النموذج، والشكل التالى يوضح شكل قائمة الأمر:



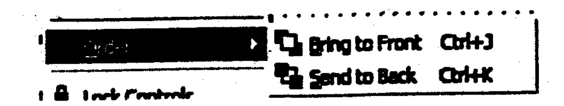
٧- القائمة الفرعية لأمر ترتيب العناصر Order :

الغرض منها:

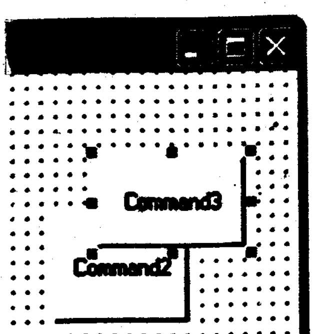
التحكم فى ظهور العنصر الحالى بالنسبة لباقى العناصر، حيث يمكـــن أن تتراكب عدة عناصر فوق بعضها البعض مما يضطر معه المصمم للحاجة إلى تقديم أو تأخير أحد العناصر ليظهر فوق أو أسفل العنصر الأخر.

التنفيذ:

تحديد بالعنصر أو العناصر المطلوب إعادة ترتيب ظهورها، ثم تنفيذ الأمر من القائمة أو من لوحة المفاتيح من خلال مفتاحى Ctrl+j لإحضار العنصـــر إلى الأمام أو Ctrl+k لإرسال العنصر للخلف، انظر الشكل التالى:



والشكل التالى يوضح أحد العناصر فى حالة تقدم على أخر :



ملاحظة : أن الر رقم ٣ تم إحضاره أمام الزر رقم ٢.

٨- أمر إغلاق العناصر Lock Controls :

الغرض منها:

منع جميع العناصر عن التحريك فى موضعها على النموذج، حيث يستخدم هذا عند وضع العنصر فى موضع معين ولا يرغب المصمم فى تعديل موضعه حاليا أثناء التصميم، أو فى حالة انتهاء التصميم المرئى للبرنامج.

التنفيذ:

تنفيذ الأمر من القائمة، ولإلغاء الإغلاق يمكن إعادة تنفيذ الأمر مرة أخرى بعد تحديد العناصر.

٢ - ٦ قائمة تصحيح الأخطاء Debug:

الغرض منها :

ويتم التعامل بشكل واسع مع هذه القائمة داخل نافذة الـــ Code

والشكل التالى يوضح شكل هذه القائمة:

Debug Run Query Diagram Tools Add-In	
Step Into	F8
Step Over	Shift+F8
Step Out	Ctrl+Shift+F8
Run To Cursor	Ctrl+F8
Add Watch...	
Edit Watch...	Ctrl+W
Quick Watch...	Shift+F9
Toggle Breakpoint	F9
Clear All Breakpoints	Ctrl+Shift+F9
Set Next Statement	Ctrl+F9
Show Next Statement	

ملاحظة :

تستخدم أوامر هذه القائمة لإجراء عمليات التعديل والتصحيح بالبرامج، وتحتاج مساحة ووقت أكثر لا تتسع مع محتويات هذا الكتاب، ويحتاجها الأكثر تخصصاً في البرمجة بهذا النظام، حاول الإطلاع علي هذه المحتويات.

٢ - ٧ قائمة تشغيل Run :

الغرض منها :

تشغيل البرنامج الذى تم تصميمه بشكل تجريبى لاختبار كود البرنامج وشــكل النموذج والارتباطات الموجودة بين نماذج أخــرى إن وجــدت ... ، أى أن ذلــك عبارة عن تجربة فعلية للتأكد من صحة تنفيذ البرنامج أثناء التصميم والبرمجة وقبل ترجمته الفعلية على ملف تنفيذى بامتداد EXE.

وهذا مماثل لأمر Run فى البيسيك العادى، وذلك لمتابعة خطــوات البرنــامج الذى تقوم بإعداده.

وبالطبع تعتبر هذه القائمة من أهم القوائم لأنها المسئولة عن التنفيذ للبرنــامج، وبالتالى إجراء عمليات التصحيح والتعديل المختلفة على البرنامج فى حالة وجــود أخطاء، وبالتالى تصحيح الأخطاء وإعادة تنفيذ البرنامج مره أخرى.

والشكل التالى يوضح شكل هذه القائمة :

Run	Query	Diagram	Tools	Add-Ins
▶ Start	F5			
Start With Full Compile	Ctrl+F5			
❚❚ Break	Ctrl+Break			
■ End				
Restart	Shift+F5			

ويلاحظ أن أوامرها الغير نشطة تنشط أثناء تشغيل البرنامج للاختبار بالضغط على الأمر Run لتكون بالشكل التالى:

Run	Query	Diagram	Tools	Add-Ins
▶ Start	F5			
Start With Full Compile	Ctrl+F5			
❚❚ Break	Ctrl+Break			
■ End				
Restart	Shift+F5			

- وفيما يلى شرح لأوامرها:

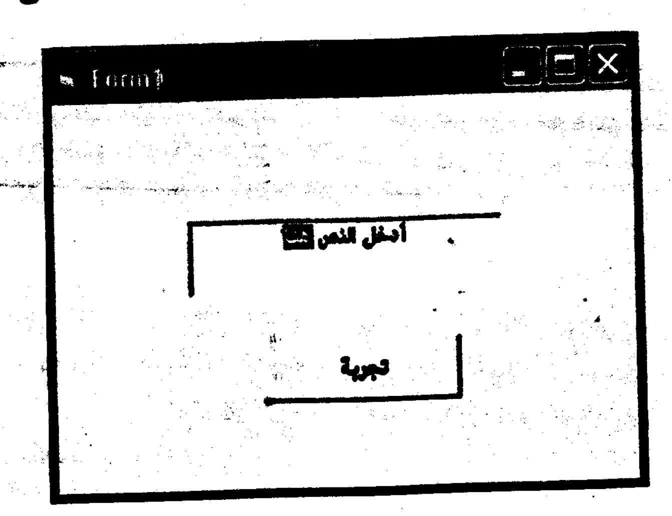
١- أمر البدء Start :

الغرض منها:

تنفيذ البرنامج للتجربة، وهو الأمر الهام الخاص ببدء التجربة أثناء التنفيذ وقبل الترجمة، وفى حالة إهمال استخدام هذا الأمر غالبا ما سيظهر للمصمم عديد وعديد من الأخطاء البرمجية والتصميمية فى برنامجه التنفيذى، لذلك فلا يمكن للمصمم الاستغناء عن هذا الأمر أثناء تصميم البرامج الكبيرة والمعقدة.

التنفيذ:

يتم التنفيذ من خلال القائمة، أو من خلال مفتاح F5 من لوحة المفاتيح، أو من خلال شريط الأدوات القياسى Standard من الرمز المسمى Start.
وعند التنفيذ لأحد البرامج مثلا يظهر شكل مقارب للشكل التالى:



ولإلغاء التنفيذ والرجوع لنافذة التصميم يتم النقر على زر الإغلاق من أعلى يمين النموذج، أو من خلال فتح القائمة واختيار الأمر End، أو من خلال الضغط على الرمز End من شريط الأدوات.

٢- أمر البدء بترجمة كاملة Start With Full Compile :

الغرض منها:

مثل الأمر السابق مع اختلاف أن هذا الأمر يقوم بترجمة جميع البرامج الفرعية بالبرنامج.

التنفيذ:

يتم التنفيذ من خلال القائمة، أو من خلال مفتاح Ctrl+F5 من لوحة المفاتيح.

- قم بالتنفيذ بنفسك.

٣- أمر التوقف المؤقت Break :

الغرض منها:

توقف مؤقت للبرنامج بغرض إضافة أو تعديل فى كود البرنامج بدلا من إغلاق وضع التشغيل والذهاب لوضع التصميم لتعديل الكود ثم التشغيل مرة أخرى، وذلك يوفر وقت المصمم لتنفيذ التعديل الذى يرغبه.

التنفيذ:

الضغط على الأمر من خلال القائمة، أو من خلال الضغط على زر Break من شريط الأدوات، ثم إعادة الاستمرار والتنفيذ مرة أخرى بالضغط على الأمر الجديد الذى يظهر بأول قائمة Run باسم Continue.

- قم بالتنفيذ بنفسك.

٤- أمر الإنهاء End :

الغرض منها:
إنهاء وضع التنفيذ والعودة لوضع التصميم مرة أخرى.

التنفيذ:
الضغط على الأمر من خلال القائمـــة، أو مـــن خـــلال الضـــغط علـــى زر End من شريط الأدوات.
- قم بالتنفيذ بنفسك.

٥- أمر إعادة التشغيل Restart:

الغرض منها:
إعادة تشغيل البرنامج أثناء توقفه المؤقت للتعديل أو بعد اكتشـــاف خطـــأ برمجى أثناء اختباره.

التنفيذ:
الضغط على الأمر من خلال القائمـــة، أو مـــن خـــلال الضـــغط علـــى مفتاحى Shift+F5 من لوحة المفاتيح.

- قم بالتنفيذ بنفسك.

٢ – ٨ قائمة الاستعلام Query :

الغرض منها :

الاستعلام عن بيانات مختلفة وإعداد تقارير لاستخدامها في عمليات الإخراج.

والشكل التالى يوضح شكل هذه القائمة:

ign]

Query Diagram Tools Add-Ins Windc

Run
Clear Results

Verify SQL Syntax

Group By
Change Type ▶

Add To Output

Sort Ascending
Sort Decending

Remove Filter

Select All Rows From <Table A>
Select All Rows From <Table B>

✍ ملاحظة :

تستخدم أوامر هذه القائمة لإجراء عمليات التشغيل وإعداد التقارير والترتيب للمحتويات بأشكال مختلفة والتحقق منها، وتحتاج مساحة ووقت أكثر لا تتسع مع محتويات هذا الكتاب، ويحتاجها الأكثر تخصصاً في البرمجة بهذا النظام، حاول الإطلاع علي هذه المحتويات.

٢ – ٩ قائمة الرسم البيانى Diagram:

الغرض منها :

التحكم في عمليات العرض للنص ولإضافة علاقات وعرضها بين الجداول، مع إجراء عمليات الاختيار والترتيب للجداول.

والشكل التالي يوضح شكل هذه القائمة:

```
iagram  Tools  Add-Ins  Wind
    New Text Annotation
    Set Text Font
    ---------------------
    Add Related Tables
    ---------------------
    Show Relationship Labels
    ---------------------
    Modify Custom View
    ---------------------
    View Page Breaks
    Recalculate Page Breaks
    ---------------------
    Arrange Selection
    Arrange Tables
    ---------------------
    Autosize Selected Tables
```

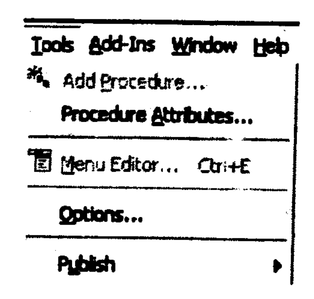
 **ملاحظة :**

قم بتنفيذ الأوامر السابقة بنفسك ، مع العلم تحتاج هذه القائمة مساحة ووقت أكثر لا تتسع مع محتويات هذا الكتاب، ويحتاجها الأكثر تخصصا في البرمجة بهذا النظام، حاول الإطلاع علي هذه المحتويات.

٢ – ١٠ قائمة الأدوات Tools:

الغرض منها :

هذه القائمة خاصة باستخدام بعض الأدوات الإضافية والوظائف المساعدة للمستخدم فى برنامج VB ، تتضح هذه الوظائف من الشكل التالى للنافذة:



– وفيما يلى شرح لأوامرها:

١ – أمر Add Procedure :

الغرض منها:

إضافة برنامج فرعى أو دالة ... داخل الكود، ويشترط حتى ينشط الأمر ويتمكن من تنفيذه أن يكون المصمم داخل صفحة الكود أولا.

التنفيذ:

فتح القائمة واختيار أمر Add Procedure، وبالتالى ستظهر على الشاشة نافذة جديدة بالشكل التالى:

Add Procedure

Name: | OK | Cancel

Type: Sub, Property, Function, Event

Scope: Public, Private

All Local variables as Statics

وبالطبع يجب عليك كتابة اسم برنامج الإجراء المراد إضافته فى المشروع الحالى المفتوح وذلك فى الصندوق Name المحدد لـــذلك فـــى الشـــكل الســـابق، بعد ذلك ستجده من ضمن نافذة كود المشروع Project بعد تحديد النوع والنطاق.

٢- أمر Procedure Attributes :

الغرض منها:

إضافة بعض الخصائص للبرامج الفرعية السابق إضافتها.

التنفيذ:

فتح القائمة واختيار الأمر منها، وبالتالى ستظهر علـــى الشاشـــة المربـــع الحوارى التالى:

Procedure Attributes

Name: | OK
Description: | Cancel
Apply
Project Help File: | Help Context ID: 0 | Advanced >>

ويمكن من الجزء المسمى Name تحديد اسم البرنامج الفرعـــى وإضـــافة الخصائص والوصف المطلوب له، ويمكن الضغط علـــى الـــزر Advanced للتحكم أكثر فى خصائص هذا البرنامج الفرعى.

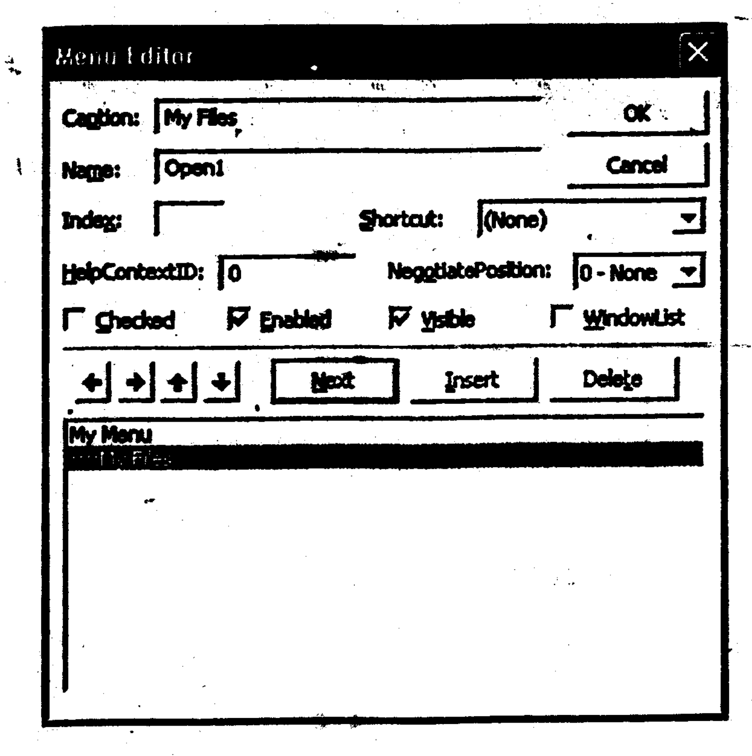
٣- أمر Menu Editor:

الغرض منها:

إضافة قوائم للبرنامج الحالى ويما تحويه من أوامر وقوائم فرعية.

التنفيذ:

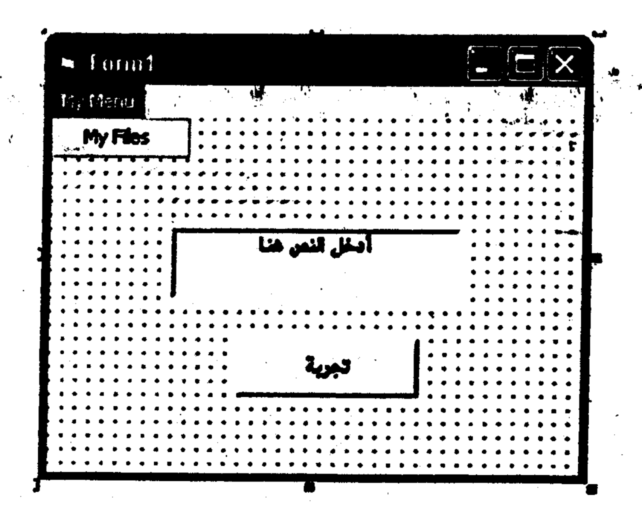
فتح القائمة واختيار الأمر منها، أو من خلال الضغط على مفتاحى Ctrl+E من لوحة المفاتيح، وبالتالى ستظهر على الشاشة المربع الحوارى التالى:



ويتم إضافة أسماء القوائم فى المربع Caption والاسم البرمجى للقائمة فى المربع Name، وبالتالى يظهر اسمها داخل الجزء الأسفل للنافذة السابقة.

ويمكن إضافة أمر داخل القائمة من خلال الإضافة بالشكل السابق ثم الضغط على الأسهم الموجودة بوسط النافذة السابق، فعند الضغط على السهم المتجه لليمين يتم إضافة العنصر المحدد كأمر لخل القائمة السابقة له فى ترتيب الظهور على النافذة السابقة.

بعد الضغط على Ok يظهر شكل مشابه للشكل التالى:



لاحظ ظهور القائمة وما بها من أوامر، وقم بالتنفيذ بنفسك.

٤- أمر Option:

الغرض منها:

التحكم فى خصائص برنامج VB العامة، مثل اختبار الكود بشكل آلى أو حجم خطوط الشبكة أو نوع الخط وحجمه للكود وغير ذلك.

التنفيذ:

فتح القائمة واختيار الأمر منها، ليظهر الشكل التالى:

Options

Editor | Editor Format | General | Docking | Environment | Advanced

Code Settings
- [x] Auto Syntax Check
- [ ] Require Variable Declaration
- [x] Auto List Members
- [x] Auto Quick Info
- [x] Auto Data Tips
- [x] Auto Indent

Tab Width: 4

Window Settings
- [x] Drag-and-Drop Text Editing
- [x] Default to Full Module View
- [x] Procedure Separator

OK | Cancel | Help

✍ ملاحظة :

قم بتنفيذ الخيارات المختلفة الموجودة بالنافذة وباقي الأوامر الخاصة بالقائمة، مع التعرف علي الوظائف الخاصة بكل منها عند التنفيذ لها.

٢ – ١١ قائمة الإضافات Add-Ins:

الغرض منها :
إمكانية إضافة قواعد بيانات للبرنامج بالإضافة للتحكم فى خدمات البرنامج من حيث التشغيل والتحميل وغيرها بالإضافة لإمكانية إعادة تحديث المكونات للبرنامج.

الشكل التالى يوضح شكل القائمة:

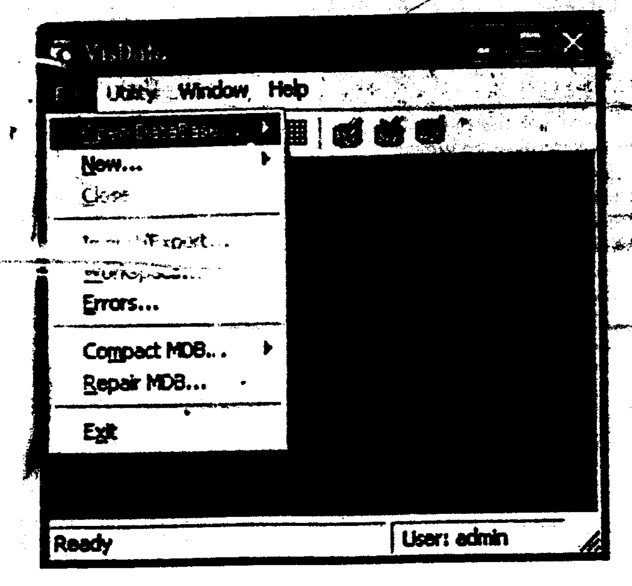
Add-Ins Window Help
Visual Data Manager...
Add-In Manager...
Component Services ▸

– وفيما يلى شرح لأوامرها:

١ – أمر Visual Data Manager :

الغرض منها:
إضافة وإنشاء والتحكم فى قواعد بيانات يريد المصمم إضافتها لبرنامجه.

التنفيذ:
اختيار الأمر من القائمة عند ذلك تظهر النافذة التالية:



ومن خلال أوامر قائمة ملف للنافذة السابقة يمكن التحكم فى فتح أو إنشـــاء أو إغلاق أو استيراد أو تصدير أو ضغط أو إصلاح قاعدة بيانات.

٢- أمر Add-Ins Manager :

الغرض منها:

للتحكم فى خدمات البرنامج من حيث التشغيل والتحميل وغيرها

التنفيذ:

اختيار الأمر من القائمة عند ذلك تظهر النافذة التالية:

Add In Manager

Available Add-Ins | Load Behavior

Startup / Loaded

DTC Framework Registrar
Package and Deployment Wizard
VB 6 ActiveX Ctrl Interface Wizard
VB 6 ActiveX Doc Migration Wizard
VB 6 Add-In Toolbar
VB 6 API Viewer
VB 6 Application Wizard
VB 6 Class Builder Utility
VB 6 Data Form Wizard
VB 6 Data Object Wizard
VB 6 Property Page Wizard

OK
Cancel
Help

Description

Load Behavior
☑ Loaded/Unloaded
☑ Load on Startup
☐ Command Line

ومن خلال النقر على الخدمة المطلوب تشـغيلها أو التعـديل بخصـائص التشغيل لها يمكن التحكم فى ذلك من الجزء السفلى الأيمـن للنافـذة السـابقة، وأخير يتم الضغط على زر Ok للحفظ والرجوع للبرنامج.

٣- أمر Component Services :

الغرض منها:

إعادة تحديث عناصر المشروع الحالى أو جميع العناصر.

التنفيذ:

اختيار الأمر من القائمة عند ذلك تظهر قائمة فرعية فيـتم اختيـار منهـا الإجراء المطلوب تنفيذه.

٢ - ١٢ قائمة إطار Window:

الغرض منها :

التحكم فى طريقة وشكل عـرض النوافـذ أمـام المصـمم علـى الشاشـة، بالإضافة لإمكانية التبديل بين نوافـذ النمـاذج والمشـروعات المفتوحـة داخـل البرنامج الحالية.

الشكل التالى يوضح شكل القائمة:

Window Help
✓ Split
Tile Horizontally
Tile Vertically
Cascade
Arrange Icons
✓ 1 Project2 - Form1 (Form)
2 Project1 - Form1 (Form)

- وفيما يلى شرح لأوامرها:

١- أمر Tile Horizontally:

الغرض منها:

ترتيب النوافذ بشكل أفقى.

٢- أمر Tile Vertically :

الغرض منها:

ترتيب النوافذ بشكل رأسى.

٣- أمر Cascade :

الغرض منها:
ترتيب النوافذ بشكل متتالى وهو الترتيب المثالى للنوافذ.

٤- أمر Arrange Icons :

الغرض منها:
ترتيب رموز النوافذ فى حالة تصغيرها.

٥- الجزء الأخير من القائمة :

الغرض منها:
يقوم بعرض أسماء النوافذ المفتوحة ومن خلال النقر عليها بالفارة يتم تنشيطها وعرضها أمام المصمم.

✍ ملاحظة :

قم بالتعرف بنفسك على ناتج التنفيذ لما سبق.

٢ – ١٣ قائمة مساعدة Help:

الغرض منها :

بالطبع مساعدة المستخدم لبرنامج Visual Basic، حيث تحتوى على كثير من الحالات الخاصة بالمساعدة والتى تتضح من شكل قائمة المساعدة Help التالية:

Help
Contents...
Index...
Search...
Technical Support
Microsoft on the Web
About Microsoft Visual Basic...

وبالطبع فى حالة فتحك لأحدى مكوناتها سيظهر لك قوائم ومكونات أخرى يمكنك اختيار ما يناسبك منها لعرض المعلومات اللازمة لك، ولكن يجب ملاحظة أن ذلك سيتم بشكل سليم فى حالة توافر نسخة MSDN وهى الخاصة بالمساعدة فى برامج Microsoft مثل VB.

ملحوظة:

قم بفتح الخيارات المختلفة وتعرف علي كل منها، وبالطبع كما نعرف تعتبر هذه القائمة من القوائم الهامة لما تحتويه من معلومات متنوعة ومصنفة عن هذا البرنامج.

٢ – ١٣ التعرف علي الأشرطة والنوافذ الخاصة بالبرنامج:

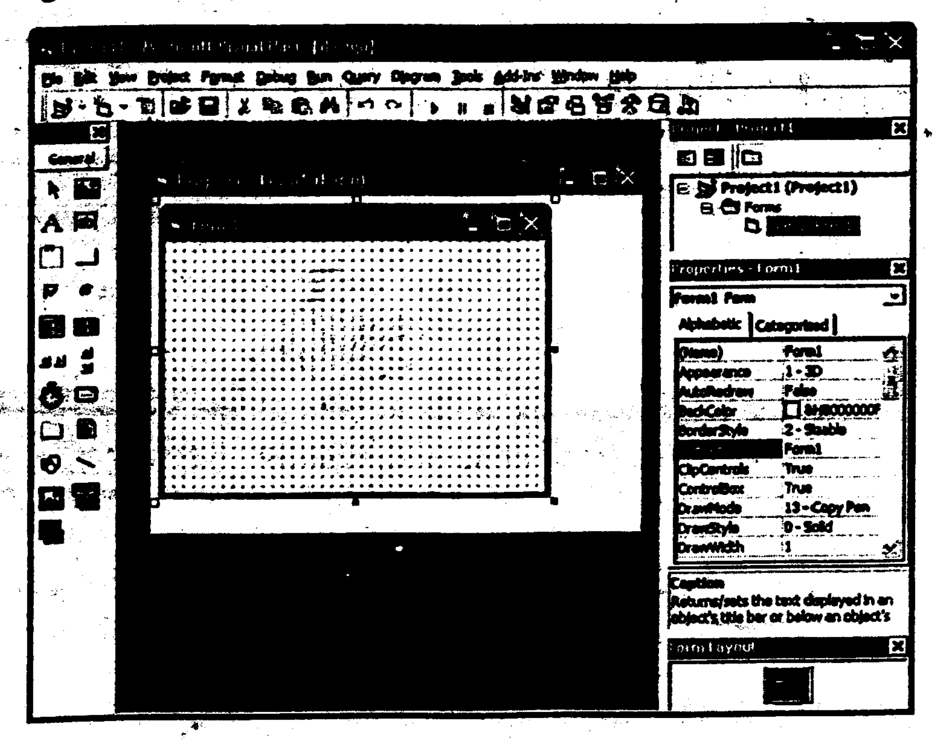
من خلال شريط القوائم Menu Bar السابق والذى يحتوى على مجموعة من الأسماء الخاصة بالقوائم الفرعية المكونة للنافذة الرئيسية التى يتم التعامل مع أوامرها أثناء إعداد وتنفيذ البرنامج .

لمكن مساعدة المستخدم للبرنامج فى التعامل مع أوامره ووظائفه المختلفة، والتى تساعد على تحقيق الهدف من تشغيل هذا البرنامج ، وهو إعداد برنامج بواسطة المستخدم لاستخدامه كتطبيق فى حل مشكلة ما بالحاسب الآلي .

مع العلم بأن هذه القوائم بالطبع قد تختلف من إصدار لأخر ، فى الغالب بالزيادة للقوائم والأوامر والتحسينات التى تدخل عليها وعلى الوظائف الخاصة بها.

شريط العنوان وشريط الأدوات
التابع لقوائم البرنامج السابقة :

لو نظرنا لشكل النافذة الرئيسية للبرنامج التى يتضح من الشكل التالى :



سنجد بالإضافة للقوائم السابقة سطرين آخرين تتوسطهما القوائم همـــا شـــريط العنوان وشريط الأدوات وفيما يلى يمكننا التعرف عليهما تكملة للاستخدام والتعرف عليهما.

رقم ١ – يمثل شريط العنوان Title Bar :

Microsoft Visual Basic [design]

الغرض منه :

بالطبع إظهار اسم أو عنوان البرنامج المستخدم للإشارة إلى وجوده أو نشاطه.
بالإضافة إلى وجود كلمة [design] بجوار Microsoft Visual Basic وتشير هذه الكلمة إلى الطور الحالى لتشغيل البرنامج، حيث قد تختلف هذه الكلمة حسب اختلاف طبيعة تشغيل البرنامج بمعنى أن هذه الكلمة تشير إلى طور تصميم البرنامج أو المشروع الآن، ويمكن أن تتغير هذه الكلمة Design إلى كلمة [Break] تشير إلى طور توقف البرنامج لإجراء أى عمليات عليه وأيضا إلى كلمة [Run] تشير إلى تشغيل أو تنفيذ البرنامج فى الوقت الحالى.

ويحتوى هذا الشريط أيضاً فى الجزء الأيمن من أعلى على ثلاث مفاتيح أو أزرار تستخدم بصفة عامة فى برامج النوافذ وهى من اليمين زر (X) وهو خاص بإغلاق البرنامج وبتنفيذه ستظهر رسالة تشير إلى ذلك وبالتالى اختيار أمر نعم أو لا أو إلغاء الأمر لتنفيذ الحالة الحالية أو النشطة، الزر الذى يليه وهو المربع وهو خاص بتكبير النافذة الحالية وملئ الشاشة بمحتوياتها وذلك للتعامل معها فقط إلى حد ما، والزر الأخير وهو الشرطة (-) يستخدم فى تصغير البرنامج الحالي ووضعه على شريط المهام لحين الحاجة إليه مرة أخرى أى بصفة مؤقتة.

رقم ٢ - شريط الأدوات Tools Bar

هناك بعض الأوامر التى يتكرر استخدامها بصفة دائمة أثناء تشغيلك للبرنامج، وبدلاً من استخدام هذه الأوامر من النوافذ السابقة يمكن أن نستدعيها من بعض الرموز الموجودة فى شريط الأدوات.
يحتوى هذا الشريط على مجموعة من الرموز أو الأيقونات التي يمكن أن تستخدم أثناء إعداد المشروع والتى يمكن استخدامها بواسطة ذوى الخبرة.

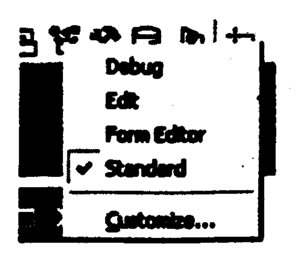
الغرض منها :

وتستخدم هذه الأدوات بغرض التسهيل على المستخدم بدلاً من تنفيذ وظائف هذه الأدوات من القوائم، ويستحسن أن تستخدم هذه الأدوات بعد معرفة المستخدم لهـا من خلال القوائم، أي بعد استخدامها من القوائم لمعرفة المستخدم لكيفيـة ونتـائج التنفيذ لها أولاً.

كيفية إظهاره :

يتم تشغيله من خلال فتح القائمة الفرعية View وتحريك مؤشر الفـأرة إلـى أسفل هذه القائمة وبالتحديد إلى أمر Toolbar والنقر عليه، المرة الأولى لتنشـيطه وظهوره فى هذا المكان (أسفل شريط أسماء القوائم).

ويمكن إخفاءه من خلال تحريك النقر بالزر الأيمن على شريط الأدوات فتظهر قائمة مختصرة يمكن من خلالها النقر على شريط الأدوات المراد إخفاءه.



يحتوى هذا الشريط على الوظائف التى يمكن تنفيذها من خلال تحريك المؤشر (سهم الفأرة) إلى الرمز المطلوب والضغط عليه وفى حالة التأكـد مـن وظيفـة الرمز يمكن تحريك المؤشر فقط على شكل الرمز وهنا سيظهر مستطيل يحتـوى

على اسم الوظيفة التى يؤديها هذه الأيقونة أو الرمز، وذلك للتأكد من الوظيفة قبـــل تنفيذها.

* وفيما يلى مجموعة من الرموز المستخدمة معه :

١ - رمز إنشاء مشروع جديد Project

- Standard EXE
- ActiveX EXE
- ActiveX DLL
- ActiveX Control

٢ - رمز الشكل أو نافذة جديدة New Form

- Form
- MDI Form
- Module
- Class Module
- User Control
- Property Page
- User Document
- WebClass
- Data Report
- DHTML Page
- Add File.... Ctrl+D

٣- رمز إنشاء قوائم المشروع Menu Editor

٤ - رمز فتح مشروع جديد Open Project
٥ - رمز حفظ المشروع الحالى Save Project
٦ - رمز غلق التحكمات Lock Controls

٧ - رمز مستعرض عناصر المشروع Project Explorer

٨ - رمز نافذة الخصائص Properties
٩ - رمز ضبط موضع النموذج Form Layout Window
١٠ - رمز متصفح الكائنات Object Browser
١١ - رمز عرض صندوق الأدوات ToolBox
١٢ - رمز شاشة عرض البيانات Data View Window
١٣ - رمز إدارة المكونات Visual Component Manager

١٤ - رمز البدء فى تشغيل البرنامج (الترجمة) Start

١٥ - رمز التوقف للبرنامج Break
١٦ - رمز إنهاء البرنامج النشط End

١٧ - رمز وضع أو حذف نقطة توقف Toggle Break Point
١٨ - رمز مراقبة متغير Instant Watch
١٩ - رمز عرض الاستدعاءات Calls
٢٠ - رمز تنفيذ البرنامج خطوه خطوة Step Into
٢١ - رمز تنفيذ البرنامج إجراء Step Over

من خلال عرض نافذة البرنامج الرئيسية السابقة يتضح أنها تحتوى على النوافذ الرئيسية التالية الخاصة بتشغيل البرنامج ككل والتى يجب على المستخدم التعـرف عليها، وفيما يلى شرح مفصل لكل منها على حده (بالفصل التالي) :

١ – نافذة البرنامج (التصميم) Form

٢ – نافذة صندوق الأدوات Tool Box

٣ – نافذة الخصائص Properties

٤ – نافذة المشروع Project

٥ – نافذة الكـود (برنامج البيسيك) Code

٦– نافذة التحكم فى موضع النموذج Layout

٧ – نافذة تحكم الألوان Color Palette

الفصل الثالث

التعامل مع نافذة Form
وصندوق الأدوات Toolsbox

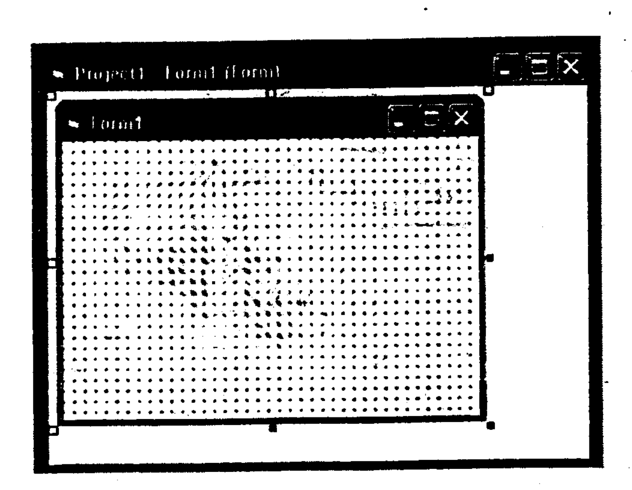
يتناول هذا الفصل المحتويات التالية :

٣ - ١ نافذة البرنامج (الشكل أو التصميم) Form

هذه النافذة تعتبر الوعاء أو القالب الذي يتم تصميم أو وضع جميع أو اغلب الأدوات الخاصة بنافذة الأدوات عليه، فهذه الأدوات (أدوات الـ VB) لا يمكن أن تستخدم وحدها بل يجب أن تكون فوق نافذة البرنامج Form.

وبالطبع يمكن أن تكون للبرنامج أكثر من نافذة وذلك حسب حاجة برنامجك.

والشكل التالى يوضح شكل النافذة :



ملاحظة :

أن هذه النافذة تحتوى على نقاط كثيرة بطريقة منتظمة والغرض من وجودها تحديد المواقع التى يمكن الرسم أو وضع الأدوات عليها، وذلك تسهيلا على مصمم نافذة البرنامج فى عمل التصميم السليم.

على سبيل المثال :

يمكن البدء في وضع أدوات معينة خاص بالنصوص الثابتة عند بدايـــة نقطـــة معينة على نافذة البرنامج.

ويلاحظ أن هذه النقاط تختفي في موقع الدالة المصممة علـــى شـــكل النافـــذة، وأيضاً تختفي جميع هذه النقاط من على شكل نافذة البرنامج وذلك في طور تنفيـــذ البرنامج من خلال قائمة التشغيل Run لبرنامج المشروع.

الغرض منها :

تصميم شكل قوائم البرنامج وشكل شاشات الإدخال للبيانات أو شاشات الإخراج للمعلومات والنتائج، وبالتالي مساعدة المستخدم (مخطط البرنامج) فـــي تصميم شكل أو شاشات البرنامج الخاص به بسهولة ويسر.

وبالطبع هذه الطريقة تمتاز بتوفير كثير من الوقت والمجهود عن طرق تصميم البرامج العادية أو التي يتم تصميمها في اللغات والأنظمة التقليدية، حيث تحتاج تصميم برنامج الإدخال الخاص بمشكلة ما إلى كتابة مجموعة من الأوامر بطريقة معينة لأداء هذه الوظيفة، بينما يمكن إعداد شاشة برنامج الإدخال بســـهولة ويســـر وبدون كتابة أوامر كثيرة بواسطة برنامج Visual Basic وبالتحديد فـــي نافـــذة التصميم أو الشكل (Form).

* كيفية إظهارها:

يتم إظهارها بطريقة افتراضيه ومباشرة عند تشغيل أو بعد تنفيذ أمر مشـــروع جديد New Project من القائمة الأولى الفرعية File وذلك بتحريك مؤشر الفأرة إلى قائمة ملف واختيار أمر New Project.

أو من خلال شريط الأدوات بالضغط على رمز الأداة الخاصة بإضافة مشروع جديد Add Standard EXE Project

ولإضافة نموذج Form جديد يمكن ذلك من خلال الرمــز Add form مــن شريط الأدوات.

وبالتالي يمكن تصميم شكل الشاشة المطلوبة فى عمليات الإدخال أو الإخــراج وذلك باستخدام نافذة صندوق الأدوات ومؤشر الفأرة ولوحة المفاتيح لإعداد الشكل المطلوب ببساطة وسهولة وأيضا سرعة فى التنفيذ. وبــالطبع لا غنــى لمســتخدم برنامج Visual Basic عن هذه النافذة.

وبفرض أنك أنشأت نافذة واحدة فستأخذ اسم Form1، وبعد ذلك أردت أن تنشأ نافذة أخرى فى نفس المشرع (التخطيط) بتنفيذ أمر الإنشاء Add Form، فهنــا سيتم إعطاء اسم جديد لهذه النافذة وهو نموذج ٢ أو Form2 وهكذا بالنسبة للنوافذ الجديدة.

✍ ملاحظة :

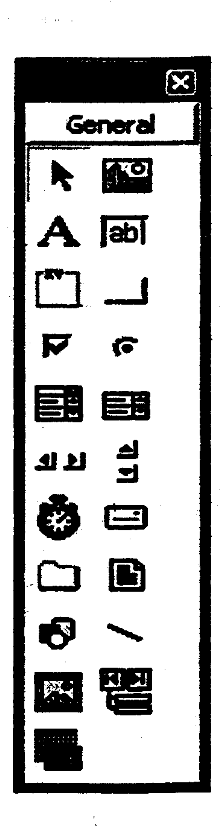
١- يمكن تغيير اسم نافذة الــ Form من خلال أمر الــ Caption فــى نافذة الخصائص المرتبطة بهذه النافذة (Form) -وسوف يتم ذكــر وشرح أمثلة لذلك عند تنفيذ برامج فعلية فيما يلى بعد -.

٢- يمكن بالطبع التعامل مع هذه النافذة بصفة مستقلة من حيث تكبير النافذة وتصغيرها وغلقها، وذلك من خلال المفاتيح أو الأزرار الموجودة فــى أعلى النافذة على اليمين وكما هو متبع فى النوافذ الأخرى المتوافقة مع برنامج النوافذ أو المثبتة عليه.

٣ - ٢ نافذة صندوق الأدوات Tool Box

الغرض منها :

إظهار مجموعة من الأدوات التى يستخدمها مخططـى البـرامج فـى إعـداد برامجهم، وبالطبع تقوم هذه الأدوات بأداء وظائف معينة تستخدم فـى البرنـامج، والغرض منها تسهيل إعداد البرامج وتوفير كثير من التكـاليف الخاصـة بإعـداد البرامج وأيضاً تسهيل التعامل مع المستخدم عند تشغيله للبرنامج، والشكل التـالى يوضح شكل هذا الصندوق:



كيفية إظهاره :

يمكن إظهار هذا الصندوق من خلال القائمة الفرعية View، حيث يمكن فـتح القائمة كالعادة، ثم التحرك بالمؤشر إلى أمر Toolbox والنقـر عليـه، وبالتـالى سيظهر صندوق الأدوات الخاص ببرنامج Visual Basic.

ويمكن إخفاء هذا الصندوق من خلال تحريك مؤشر الفأرة على الصندوق الصغير الذى يحتوى على العلامة (X) ثم الضغط عليه بزر الفأرة الأيسر.

*** وتتكون هذه النافذة من الأدوات التالية :**

المؤشر (ليس أداة)	Pointer
صندوق أو خانة الصورة	Picture Box
العنوان	Label
خانة النص	Text Box
الإطار	Frame
مفتاح أو زر الأوامر	Command Button
خانة الاختبار	Check Box
مفتاح الاختيار	Option Button
القائمة المنسدلة	Combo Box

خانة القائمة	List Box
شريط التمرير الأفقي	HScroll Bar
صندوق التمرير الرأسي	Vscroll Bar Box
الوقت المؤقت	Timer
قائمة الوحدات	Driver List
قائمة الأدلة	DirList Box
قائمة الملفات	File List
أداة الأشكال الهندسية	Shape
أداة الخطوط	Line
أداة الرسم	Image
أداة قاعدة البيانات	Data
أداة ربط الكائنات	OLE

ملاحظة :

يمكن إضافة أدوات أخرى من خلال قائمة Project واختيار Components، وهذه الأدوات قد تختلف من برنامج لآخر (إصدار لآخر) بالزيادة، وفى الغالب بالزيادة لرموز جديدة ووظائف جديدة لم تكن مدرجة من قبل وذلك بالطبع لتحسين الأداء وزيادة قدرة البرنامج وأيضاً لسهولة التعامل معه.

وفيما يلى شرح لبعض هذه الوظائف (الأدوات) :

١ - المؤشر Pointer

هذه المؤشر أو السهم لا يعتبر أداة من الأدوات المستخدمة فى هذا البرنـــامج، ولكن يستخدم هذا المؤشر لتغيير حجم أو موقع الأدوات الأخرى المنفذة على نوافذ برنامج Visual Basic ، هذا بالإضافة إلى أنه يمثل وسيلة الإشارة إلى الأهداف أو الأدوات التى تمت إضافتها إلى نافذة الـــForm.

٢ - صندوق أو خانة الصورة Picture Box

يستخدم لوضع الصور المختلفة أو أشكال أدوات معينة بداخلة حيث يمثل هــذا الصندوق برواز لصورة ما، أو كوعاء لــلأدوات الأخـــرى ، أى يمكـــن أن يـــتم استدعاء برنامج للرسم وليكن من بيئة النوافذ مثل برنامج Paint Brush وذلـــك لوضع أو رسم صورة ما داخلة.

٣ - العنوان Label

يستخدم هذا الرمز فى عرض النصوص الثابتة على النافذة (Form) التى لا يستطيع المستخدم تعديلها ، أى لا يمكن للمستخدم تعديل هذا النص مباشرة أثنـــاء تنفيذ البرنامج، ولكنه يمكن تعديله من نافذة الكود الخاصة بتشغيله.

وبالتالى يمكن التعديل فقط فى محتواه أثناء مرحلة التصميم للبرنامج.

٤ - خانة النص Text Box

يستخدم هذا الرمز فى طلب معلومة من المستخدم أو عرض نص معين يمكـــن للمستخدم أن يعدل فيه سواء من خلال الكود أو التعديل المباشر بإدخال قيمة جديدة فيه، حيث يمكن التعديل فيه أثناء مرحلة التنفيذ.

٥ - الإطار Frame

يستخدم هذا الرمز فى رسم إطار على نافذة البرنامج لوضع مجموعة من الأدوات الأخرى التى تستخدم فى البرنامج، ولتنفيذ ذلك يمكن رسم الإطار أولاً ثم وضع الأدوات المطلوبة بداخله. حيث الضغط على الأداة المطلوبة وتحريك المؤشر إلى داخل الإطار ووضعها فى المكان المطلوب.

٦ - مفتاح أو زر الأوامر Command Button

لتصميم مفتاح أو زر على نافذة البرنامج حيث يتم التعامل معه فى مرحلة تنفيذ البرنامج، وذلك بظهور زر يستخدمه المستخدم أو يضغط على عليه، عندئذ سيتم تنفيذ مجموعة من الأوامر الموجودة به (بنافذة الكود الخاصة به) بصورة متتالية، أي يستخدم لإنشاء زر أو مفتاح أمر ينفذ مجموعة من التعليمات أو الخطوات أو الأوامر المحددة عند الضغط عليه أثناء تنفيذ البرنامج.

٧ - خانة الاختيار Check Box

يستخدم لإنشاء صندوق تساعد المستخدم على الاختيار من بين حالتين إما صحيح أو خطأ، وفى حالة الاختيار الصحيح أي النشط أو الفعال ستظهر علامة (x) فى مربع صغير بصندوق الاختيار هذا وبذلك تأخذ الخاصية القيمة True، وأما فى حالة الاختيار غير صحيح فلا يظهر علامة بهذا الصندوق وبالتالي تأخذ الخاصية False أي خطأ.

٨ - مفتاح الاختيار Option Button

يستخدم لعرض مجموعة خيارات على المستخدم أن يختار إحداها، وعندما يقف على أحد الخيارات أو يختار المستخدم أحدها ستظهر بداخل الدائرة الحالية دائرة أخرى صغيرة تشير إلى حالة القيمة المختارة صحيحة True، وبالعكس تختفي الدائرة عند الذهاب أو الوقوف على اختيار أخر من الخيارات الموجودة.

٩ - الصندوق أو الخانة المركبة Combo Box

اسم هذه الصندوق مشتق من Combination ومعناها المركب، يعطى هــذه الخانة المستخدم حرية الاختيار لتعليمات أو تنفيذ أوامر من خلال قائمة أو إدخــال اختياره كتابة. حيث يمكن على سبيل المثال اختيار عنصر القائمة لفــتح ملــف أو حفظ ملف أو إغلاق ملف، حيث يمكن أن نكتب اسم الملف أو نختاره من القائمـــة. وعند زيادة محتويات هذا الصندوق (عناصره) عن إرتفاع الصــندوق ســيظهر قضبان الانزلاق الرأسية (الأسهم) على جانبي الصندوق وبالطبع يمكن التعامــل معها لتنفيذ الاختيار المطلوب من خلال مؤشر الفأرة.

١٠ - خانة القائمة List Box

تستخدم لعرض قائمة من العناصر مرتبة أو غير مرتبة لكى يتم اختيار أحدها حيث يمكن أن يحتوى هذه الخانة أو الصندوق على عدة أعمــدة ، وعنــد زيــادة محتويات هذا الصندوق (عناصره) عن عــرض الصــندوق ســيظهر قضــبان الانزلاق الرأسية (الأسهم) على جانب الصندوق الأيمن وبالطبع يمكن التعامــل معها لتنفيذ الاختيار المطلوب من خلال مؤشر الفأرة.

١١ - شريط التمرير الأفقي HScroll Bar

معناه شريط أو قضيب التمرير أو الانزلاق الأفقي Horizontal Scroll Bar يستخدم هذا الشريط للتسهيل على المستخدم فى عرض محتويات قائمة ما أو نافذة ما أو الحصول على معلومات من المستخدم، حيث يتحرك هذا الشــريط يمينــاً أو يساراً حسب حركة المستخدم. بمعنى أنه يعتبر أداه سريعة للمستخدم تمكنــه مــن الانتقال أفقيا فى نافذة ما تحتوى عناصر أو بيانات كثيرة. وأيضا يفيد فى عمليــة تحديد موقع أو مكان الوقوف على البيانات حيث سيظهر مربع صغير يوضح هــذا الموقع.

١٢ - صندوق التمرير الرأسي Vscroll Bar Box

معناه شريط أو قضيب التمرير أو الانـزلاق الأفقـي Vertical Scroll Bar يستخدم هذا الشريط للتسهيل على المستخدم فى عرض محتويات قائمة ما أو نافذة ما أو الحصول على معلومات من المستخدم، حيث يتحرك هذا الشريط لأعلـى أو لأسفل حسب حركة المستخدم. بمعنى أنه يعتبر أداه سريعة للمستخدم تمكنه مـن الانتقال رأسياً فى نافذة ما تحتوى عناصر أو بيانات كثيرة. وأيضا يفيد فى عمليـة تحديد موقع أو مكان الوقوف على البيانات حيث سيظهر مربع صغير يوضح هـذا الموقع.

١٣ - الوقت (المؤقت) Timer

يشير هذا الرمز أو هذه الأيقونة أو الأداة إلى تنفيذ مجموعة من الأوامر كـل فترة زمنية محددة. وهذه الأداة لا تكون ظاهرة فى وقت تنفيذ البرنامج ولكن يـتم تنفيذها بطريقة آلية.

١٤ - قائمة الوحدات Driver List

من خلال هذه الأداة يمكن عرض قائمة وحدات الإدارة الملحقة أو الموجـودة بالجهاز، وبالتالى يمكن التعامل معها بالاختيار من خلال هذا الصـندوق للوحـدة المراد التعامل معها.

١٥ - قائمة الأدلة DirList Box

من خلال هذه الأداة يمكن عرض قائمة بالأدلة أو الفهـارس الموجـودة فـى الجهاز وبالتحديد وحدة الإدارة الحالية النشطة بالجهاز وبالتالي يمكن التعامل معها بالاختيار من خلال هذا الصندوق.

١٦ - قائمة الملفات File List

من خلال هذه الأداة يمكن عرض قائمة بالملفات الموجودة على الوحدة الحالية أو الاسطوانة الحالية وأيضا من على الأدلة أو الفهارس الموجودة في الجهاز وبالتحديد وحدة الإدارة الحالية النشطة بالجهاز وبالتالي يمكن التعامل معها بالاختيار لها من خلال هذا الصندوق.

١٧ - أداة الأشكال الهندسية Shape

من خلال هذه الأداة يمكن وضع أشكال هندسية مثل الدوائر والمستطيلات والمربعات والشكل البيضاوى.. الخ فى نافذة البرنامج Form أثناء التصميم لها.

١٨ - أداة الخطوط Line

من خلال هذه الأداة يمكن رسم أكثر من شكل للخط وذلك أيضا أثناء تصميم البرنامج.

١٩ - أداة الرسم Image

تستخدم هذه الأداة لعرض صورة أو أيقونه فى نافذة البرنامج وذلك لتزيين هذه الشاشة أو النافذة أو واجهة البرنامج أثناء التنفيذ له.

٢٠ - أداة قاعدة البيانات Data

تستخدم هذه الأداة للدخول إلى ملفات قواعد البيانات المرفقة أو المستخدمة أثناء التطبيق للتعامل معها.

٢١ - أداة ربط الكائنات OLE

هذا الاسم مشتق من Object Linking and Embedding لتشير إلى الربط والتضمين للكائنات الأخرى من التطبيقات الأخرى التى تعمل مع هذا البرنامج. مثل نقل صورة من تطبيق خاص بالرسم إلى برنامج Visual Basic حتى يمكن التعديل بها فى الرسم الذى تم تضمينه فى البرنامج.

٣ - ٣ مراحل التعامل مع (تشغيل) الأدوات :

أولاً : كيفية وضع الأدوات على نافذة البرنامج Form

يمكن تنفيذ ذلك من خلال الخطوات التالية :

- تحريك مؤشر الفأرة إلى رمز الأداة المطلوبة.
- الضغط أو النقر عليها باستخدام الزر الأيسر للفأرة مرة واحدة.
- عندئذ سنلاحظ أن رمز الأداة المختارة مضغوط إلى أسفل.
- بالتالى يمكن تحريك مؤشر الفـأرة إلـى نافـذة البرنـامج Form والـذهاب للموقع الذى تريده لها وسنجد أن شكل المؤشر قد تغير على نافذة الـ Form إلى شكل علامة (+).
- عندئذ يمكن الضغط على الزر الأيسر للفأرة مع السحب لليسار ريـثما تـنتهى لتحديد موقعها وحجمهـا أو الضـغط علـى الـزر الأيسـر مـع السـحب لأى اتجاه آخر، ثم ترك الزر الأيسر للفأرة.
- عندئذ ستنتقل الأداة أو شكل الأداة إلى نافذة البرنامج Form وتثبـت عليـه، وهكذا بالنسبة لأى أداة أخرى.

على سبيل المثال :

فى حالة استخدام أداة عمل عنوان Label (أى عمل نص ثابت على نافـذة الـ Form)كأن نكتب عنوان قائمة ما أو عنصر قائمة مـا، وهـذا العنصـر أو النص لا يتغير محتواه أثناء تنفيذ البرنامج ولكن يمكن تغييره من خلال كود معين.

مثل : كتابة عبارة " مرحلة الإدخال للبيانات " للإشارة إلى تشـغيل برنـامج إدخال البيانات.

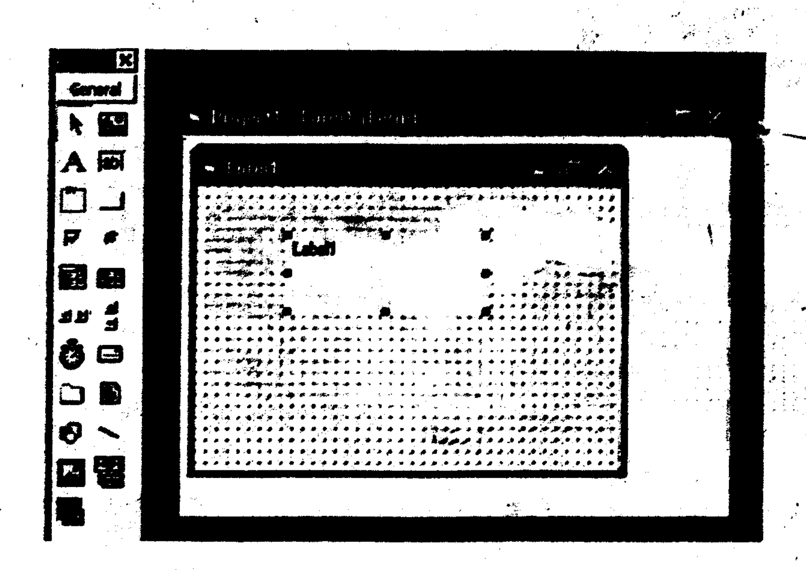
• لتنفيذ ذلك يجب إتباع الآتي :

- بفرض وجود نافذة الـــ Form جاهزة ولكن Form1.

- الذهاب إلى صندوق الأدوات والتحرك بالمؤشر إلى الرمز الذى يمكن من خلاله كتابة النص الثابت وهو رمز Label المرسوم عليه حرف (A).

- النقر عليه مرة واحدة بزر الفأرة الأيسر، وبالطبع سيظهر شـــكل الأداة وكأنـــه مضغوط إلى أسفل للإشارة إلى بداية التنفيذ.

- التحرك بالمؤشر إلى نافذة الـــ Form ولصق أو تثبيت رمز الأداة على النافـــذة كما قلنا من قبل بالذهاب إلى الموقع المحدد لها والضغط على الزر الأيسر بالفأرة مع السحب لرسم الأداة إلى أى اتجاه ثم ترك الزر الأيسر وبالتالى ستظهر النافذة بالشكل التالى:



MR

- عندئذ يمكن الكتابة مباشرة من خلال لوحة المفاتيح للنص الثابت المراد وضعه على النافذة وذلك من خلال الخاصية المسماة Caption للأداة الحالية وليكن كما قلنا عبارة " مرحلة الإدخال للبيانات ".
- ثم الضغط على الزر الأيسر للفارة لحفظ أو تثبيت العبارة على شكل الـForm1.
- مع ملاحظة أن صندوق هذه الدالة محاط بنقاط داكنة للتحكم في حجم الدالة على نافذة البرنامج، حيث يمكن تحريك المؤشر على هذه النقاط وسيظهر عليها رمز له شكل السهم من طرفيه، معنى ذلك أنه يمكن تحريك هذا السهم مع تثبيت الزر الأيسر للفأرة حسب الاتجاه لضبط الحجم، ثم ترك الزر الأيسر لتثبيت الحجم.
- وأيضا يمكن أن يتم التعديل في النص بعد ذلك خلال نافذة الخصائص أيضاً، حيث يظهر النص الثابت الذي يتم كتابته على نافذة الخصائص عند أمر Caption مع ظهور مؤشر على شكل " | " يمكن تحريكه إلى مكان الخطأ وإجراء التعديل عليه، لاحظ الشكل التالي:

Properties - Label1

Label1 Label

Alphabetic | Categorized

AutoSize	False
BackColor	&H8000000F&
BackStyle	1 - Opaque
BorderStyle	0 - None
Caption	مرحلة الإدخال للبيانات
DataField	
DataFormat	
DataMember	
DataSource	
DragIcon	(None)
DragMode	0 - Manual



Caption
Returns/sets the text displayed in an object's title

وهكذا بالنسبة لأي أداة أخرى، ثم متابعة الإجراءات الأخرى الخاصة بعملية الحفظ لنافذة البرنامج والمشروع ... الخ.

٣ - ٤ كيفية تغيير مواقع الأدوات :

يتضح من المرحلة السابقة أنه تم تحديد أو تثبيت موقع لهذه الأداة على نافذة البرنامج Form1، ولكن يمكن التحكم فى هذا الموقع بتغييره على أى موقع أخر على نافذة البرنامج وذلك من خلال الطرق الآتية:

١ - طريقة السحب والإلقاء Drag and Drop :

- تحريك المؤشر على صندوق الدالة المثبت على نافذة الـ Form والذى يحتوى على عبارة "مرحلة الإدخال لبيانات ".
- الضغط عليه ضغطة واحدة وهنا سيظهر النقاط الداكنة محاطة بهذا الصندوق.
- وبالتالى يمكن تحريك مؤشر الفأرة إلى أى موقع بهذا الصندوق.
- ثم يتم الضغط على الزر الأيسر للفأرة مع السحب لأى موقع تريده على نافذة البرنامج، ثم ترك الزر الأيسر.
- عندئذ ستثبت الدالة فى هذا الموقع الجديد ومحاط أيضاً بصندوقها النقاط الداكنة للإشارة إلى أننا مازلنا فى طور التعديل.
- لذا يمكن تحريك مؤشر الفأرة إلى خارج هذا الصندوق وبالتالى ستختفي هذه النقاط وبالتالى سيتم التثبيت لهذا التعديل.

٢ - طريقة التحديد بالمقياس :

هذه الطريقة قد تكون أدق من الطريقة السابقة ولكنها تحتاج وقت أطول لتنفيذها، يتم تنفيذها من خلال نافذة الخصائص Properties ، وهذه الخاصية تستخدم مقياس (Twip) التويب وهو يساوى ١/ ١٤٤٠ من البوصة أو يساوى ١ / ٥٦٧ من السنتمتر.

وهناك خاصيتين للتعامل مع هذه الطريقة وهما Top وتستخدم لتحريك أو نقل الأداة بطريقة رأسية (أي من أعلى إلى أسفل) والخاصية الثانية Left وتستخدم لتحريك أو نقل الأداة بطريقة أفقية (أي من اليمين إلى اليسار).

ولتنفيذ ذلك يمكن إتباع الآتي :

- تنشيط نافذة الخصائص كما نعرف من خلال فتح نافذة المشاهدة View أو بالضغط على مفتاح F4.
- التحرك داخل هذه النافذة من خلال شريط التمرير الموجود على يسار هذه القائمة والذهاب إلى أمر أو خاصية Left أو أمر أو خاصية Top.
- عندئذ سيظهر مجاور للخاصية المختارة رقم وهو مقياس الموقع الحالي للأداة، وبالتالي يمكن تحريك مؤشر الفأرة عليه وكتابة رقم المقياس الذي تحدده، ثم الضغط على مفتاح Enter.
- عندئذ ستجد الأداة الموجودة على نافذة البرنامج قد تحركت حسب المقياس المحدد لها في الخاصية المحددة في نافذة الخصائص.

وهذه الطريقة تعتبر أدق في حالة رسم شاشات الإدخال أو الإخراج أو القوائم الخاصة بالبرنامج عندما ترغب في محاذاة الأدوات المستخدمة أفقيا أو رأسيا، ولكن تحتاج تدريب من المستخدم لمعرفة قياس المواقع على نافذة البرنامج (Form).

٥ - ٣ كيفية نسخ الأدوات

بنفس الأسلوب المستخدم في البرامج التي تعمل في بيئة النوافذ مثل برنامج الكتابة Word، يمكن استخدام إحدى الطرق أو الوظائف التالية لتنفيذ نسخ أداة معينة :

١ – استخدام قائمة تحرير Edit من قائمة البرنامج الرئيسية.

ولتنفيذ هذه الطريقة نستخدم أمر Cut من هذه النافذة، ويتم التنفيذ من خلال الخطوات التالية :

- تنشيط الأداة المراد قصها من على نافذة البرنامج، وذلك بتحريك مؤشر الفأرة عليها.
- ثم الضغط عليها مرة واحدة لتحديدها بظهور النقاط الداكنة على أطراف الصندوق الخاص بها.
- عندئذ يمكن تحريك مؤشر الفأرة إلى قائمة View وفتح القائمة.
- التحرك لأمر Cut والضغط عليه مرة واحدة.
- عندئذ ستلغى الأداة من على نافذة البرنامج، مع العلم أنها ما زالت فى Buffer (حافظة برنامج Windows بالذاكرة) أى يمكن استرجاعها مرة أخرى من خلال فتح نافذة البرنامج مرة أخرى والضغط على أمر لصق Paste.

٢ – استخدام المفاتيح من لوحة المفاتيح.

ولتنفيذ هذه الطريقة نستخدم مفتاح Ctrl + X ، ويتم التنفيذ من خلال الخطوات التالية :

- تنشيط الدالة المراد قصها من على نافذة البرنامج، وذلك بتحريك مؤشر الفأرة عليها.
- ثم الضغط عليها مرة واحدة لتحدد بظهور النقاط الداكنة على أطراف الصندوق الخاص بها.
- عندئذ يمكن الضغط على مفتاحى Ctrl + X، عندئذ ستلغى الدالة من على نافذة البرنامج، مع العلم أنها ما زالت فى Buffer الذاكرة أى يمكن استرجاعها مرة أخر من خلال الضغط على مفتاحى Ctrl + V.

٣ – ٦ كيفية نسخ الأدوات :

بنفس الأسلوب المستخدم فى البرامج التى تعمل فى بيئة النوافذ مثل برنامج الكتابة Word، يمكن استخدام إحدى الطرق أو الوظائف التالية لتنفيذ نسخ أداة معينة :

١ – استخدام قائمة تحرير Edit من قائمة البرنامج الرئيسية.

ولتنفيذ هذه الطريقة نستخدم أمر Copy من هذه النافذة، ويتم التنفيذ من خلال الخطوات التالية :

- تنشيط الأداة المراد نسخها من على نافذة البرنامج، وذلك بتحريك مؤشر الفأرة عليها.
- ثم الضغط عليها مرة واحدة لتحدد بظهور النقاط الداكنة على أطراف الصندوق الخاص بها.
- عندئذ يمكن تحريك مؤشر الفأرة إلى قائمة View وفتح القائمة.
- التحرك لأمر Copy والضغط عليه مرة واحدة.
- عندئذ سيتم نسخ الأداة فى الذاكرة Buffer، مع العلم أنها ستبقى فى Buffer الذاكرة، أى يمكن استرجاعها مرة أخرى، بشرط أن لا يتم تنفيذ هذا الأمر مرة أخرى إلا بعد لصقها، لأن اللصق يتم لأخر عمليه منفذة فى الذاكرة والعملية السابقة تلغى من الذاكرة المؤقتة المرتبطة بهذه العملية.
- عندئذ يمكن فتح نافذة View مرة أخرى والضغط على أمر لصق Paste.
- وهنا سيتم لصق محتوى الذاكرة إلى نافذة البرنامج النشطة أو الحالية فى الركن الأعلى من يسار النافذة، وبالتالى يمكن التعامل معها كما تريد.







٢ - استخدام المفاتيح من لوحة المفاتيح.

ولتنفيذ هذه الطريقة نستخدم مفتاحى Ctrl + c ثم مفتاحى Ctrl + V ، ويتم التنفيذ من خلال الخطوات التالية :

- تنشيط الأداة المراد نسخها من على نافذة البرنامج، وذلك بتحريك مؤشر الفأرة عليها.
- ثم الضغط عليها مرة واحدة لتحدد بظهور النقاط الداكنة على أطراف الصندوق الخاص بها.
- عندئذ يمكن الضغط على مفتاحى Ctrl + C، عندئذ ستنسخ الأداة من على نافذة البرنامج إلى الذاكرة المؤقتة.
- وبالتالى يمكن لصقها إلى أى نافذة برنامج بعد استرجاعها مرة أخر من خلال الضغط على مفتاحى Ctrl + V على نافذة البرنامج النشطة.

٣ - ٧ كيفية حذف الأدوات :

بنفس الأسلوب المستخدم فى البرامج التى تعمل فى بيئة النوافذ مثل برنامج الكتابة Word، يمكن استخدام إحدى الطرق أو الوظائف التالية لتنفيذ إلغاء أداة معينة :

١ - استخدام قائمة تحرير Edit من قائمة البرنامج الرئيسية.

ولتنفيذ هذه الطريقة نستخدم أمر Delete أو Clear من هذه النافذة، ويتم التنفيذ من خلال الخطوات التالية :

- تنشيط الأداة المراد إلغاؤها من على نافذة البرنامج، وذلك بتحريك مؤشر الفأرة عليها.
- ثم الضغط عليها مرة واحدة لتحدد بظهور النقاط الداكنة على أطراف الصندوق الخاص بها.
- عندئذ يمكن تحريك مؤشر الفأرة إلى قائمة View وفتح القائمة.
- التحرك لأمر Delete والضغط عليه مرة واحدة.
- عندئذ سيتم إلغاء الأداة من نافذة البرنامج، مع العلم أنها لا تبقى في حافظة برنامج النوافذ Buffer أى لا يمكن استرجاعها مرة.

٢ - استخدام المفاتيح من لوحة المفاتيح.

ولتنفيذ هذه الطريقة نستخدم مفتاح الإلغاء Del ، ويتم التنفيذ من خلال الخطوات التالية :

- تنشيط الأداة المراد نسخها من على نافذة البرنامج، وذلك بتحريك مؤشر الفأرة عليها.
- ثم الضغط عليها مرة واحدة لتحدد بظهور النقاط الداكنة على أطراف الصندوق الخاص بها.
- عندئذ يمكن الضغط على مفتاح Del، عندئذ ستلغى الأداة من على نافذة البرنامج.

الفصل الرابع

نافذة الخصائص Properties

يتناول هذا الفصل المحتويات التالية:

- الغرض منها
- كيفية التشغيل
- مكونات النافذة
- ضبط الخصائص
- شرح بعض الخصائص

٤ - ١ الغرض منها :

تحديد مظهر وشكل وسلوك الأدوات التى تستخدم فى برنامج VB، وهذه الخصائص تشير إلى قيم معينة ترتبط بوظائف أو أوامر كل الأداة، وأيضاً هذه القيم يستطيع معد البرنامج أن يغيرها إذا أراد.

حيث توجد لكل أداة من الأدوات المستخدمة فى برنامج VB خصائص معينة تحدد مظهر وسلوك هذه الأداة أثناء تنفيذ البرنامج المستخدمة فيه.

مع العلم بأن هذه الخصائص تعطى من برنامج VB بصفة افتراضية يعطيها البرنامج لهذه الأدوات، أي تكون هذه الخصائص ثابتة حتى يغيرها معد البرنامج طالما يريد وحسب احتياجات برنامجه.

٤ - ٢ كيفية إظهارها :

من خلال نافذة المشاهدة View يمكن اختيار أمر Properties بتحريك المؤشر عليه والضغط على Enter أو من خلال لوحة المفاتيح يمكن مباشرة الضغط على مفتاح F4، وبالتالى ستظهر نافذة الخصائص فى الغالب على اليمين من أسفل الشاشة.

وبالطبع يمكن التحكم فى وجودها على الشاشة وتحريكها مثلاً من موقعها إلى موقع آخر وذلك كعادة كل النوافذ فى برنامج النوافذ Windows بتحريك مؤشر الفأرة على سطر العنوان الخاص بها (ذو اللون الأزرق فى الغالب) الموجود أعلى النافذة والضغط على الزر الأيسر مع السحب للمكان الذى تريده ثم ترك الزر الأيسر للفأرة.

وأيضا يمكن التحكم فيها بالتكبير والتصغير والغلق لها كعادة كل النوافذ الأخرى من خلال العلامات التالية (X).

وفيما يلى شكل هذه النافذة :

نافذة الخصائص Properties

Properties - Label1

Label1 Label

Alphabetic | Categorized

(Name)	Label1
Alignment	0 - Left Justify
Appearance	1 - 3D
AutoSize	False
BackColor	&H8000000F&
BackStyle	1 - Opaque
BorderStyle	0 - None
Caption	مرحلة الإدخال للبيانات
DataField	
DataFormat	
DataMember	
DataSource	
DragIcon	(None)
DragMode	0 - Manual
Enabled	True
Font	MS Sans Serif
ForeColor	&H80000012&
Height	735
Index	
Left	1080
LinkItem	
LinkMode	0 - None
LinkTimeout	50
LinkTopic	
MouseIcon	(None)
MousePointer	0 - Default
OLEDropMode	0 - None
RightToLeft	False
TabIndex	0
Tag	
ToolTipText	
Top	600
UseMnemonic	True
Visible	True

Font
Returns a Font object.

وعند الضغط على الثلاث نقاط بالزر المجاور:

على سبيل المثال :

خاصية اسم الخط Font الغرض منها تحديد اسم الخط الذى تستخدمه الأداة للكتابة به على نافذة البرنامج Form. وذلك من حيث لختيار نوع الخط نفسه (Sinplified Arabic - Mudir - MS Senif) ونمط الخط (عادى – مائل – أسود عريض – أسود عرض مائل) وأيضا حجم الخط (٨ – ١٠ – ١٢ –..)....الخ.

ملاحظة :

هناك بعض الخصائص لبعض الأدوات يمكن أن يتم عملها أو التحكم فيها أثناء تصميم البرنامج وأيضا أثناء تنفيذ البرنامج فقط أو يمكن فقط التحكم فيها أثناء تنفيذ أو تشغيل البرنامج فقط، أى قد تكون هذه الخصائص متاحة أثناء تصميم البرنامج أو أثناء تنفيذ البرنامج.

٤ - ٣ مكونات نافذة الخصائص :

بصفة عامة تتكون النافذة من الأقسام الرئيسية التالية :

١ - خانة اسم النافذة.
٢ - خانة اسم الأداة.
٣ - خانة قائمة الخصائص.
٤ - خانة إدخال القيمة.

راجع شكل النافذة السابق.

١ - خانة اسم النافذة

تتكون هذه النافذة من الإطار العام والسطر الأعلى منها والذي يوجد عليه كلمة Properties وبجوارها اسم نافذة البرنامج النشطة وليكن Form1، ثم رموز الغلق والتكبير والتصغير ثم العناصر الرئيسية التالية.

٢ - خانة اسم الأداة :

وهو خاص بعرض لأسماء الأدوات الملحقة بنافذة البرنامج Form1، وتوجد هذه الخانة في السطر الذي يلي سطر اسم نافذة البرنامج أو أسفله ويسمى سطر اسم الأداة النشطة الآن.

وأيضا بجواره شريط تمرير رأسي متجه إلى أسفل وذلك (لفتح أو عرض الأدوات الأخرى التي توجد أيضا على نافذة البرنامج Form1، وبالطبع يمكن تحريك مؤشر الفأرة عليه والضغط عليه لفتحة والتعامل مع الأدوات التي توجد به).

٣ - قائمة الخصائص

يوجد على إطار آخر لنفس النافذة أسفل الإطار السابق من اليمين شريط تمرير رأسي وذلك لعرض جميع الخصائص والوظائف المدرجة بهذه النافذة والتي يمكن التعامل معها كما سيلي.

وتوجد هذه الخصائص فى الجانب الأيسر من النافذة.

وفيما يلى سرد لجميع الخصائص المدرجة ضمن برنامج Visual Basic Ver. 6.0 من خلال نافذة الخصائص الخاصة بالبرنامج.

✍ ملاحظة :

محتويات هذه النوافذ من الخصائص قد تختلف من أداة لأخرى أو من نافذة لأخرى أثناء مراحل التصميم أو التنفيذ لها.

٤ - خانة إدخال القيمة :

وتوجد هذه الخانة على الجانب الأيمن من النافذة، وهى مرتبطة بكل خاصية ليتم كتابة أو إدخال البيانات الجديدة الخاصة بالخاصية.

✍ ملاحظة :

فى هذه الخانة قد نجد أشكال مختلفة علي إطارها مثل مربع يوجد به سهم متجه إلى أسفل وهذا معناه أنه يوجد قائمة منسدلة بها بعض الخيارات سيتم الالتزام بها أثناء مرحلة الإدخال (أى ستحرك مؤشر الفأرة على الاختيار المناسب لك وتختاره كقيمة للخاصية)، مثل خاصية وحدات الإدارة أو خاصية الخط.

وقد يظهر مربع أخر لخصائص أخرى به مجموعة من النقاط بطريقة أفقية، وعند الضغط عليه سيظهر مربع حوارى يحتوى على القيم الخاصة بهذه الخاصية وبالطبع يمكنك اختيار ما يناسبك من القيم، مثل خاصية اللون أو خاصية DragIcon أو خاصية MouseIcon.

٤ - ٤ طرق ضبط الخصائص :

والمقصود من ضبط الخصائص هو تحديد الخصائص المناسبة للأداة المختارة، وذلك بتغيير خصائصها التي يمنحها برنامج VB عند تشغيله بما يناسب حاجات معد البرنامج لبرنامجه.

يمكن ضبط الخصائص للأدوات المختلفة الموجودة بهذا البرنامج من خلال إحدى الطريقتين الآتيتين :

١ - ضبط الخصائص أثناء تصميم البرنامج :

بمعنى أن عملية تحديد الخصائص تتم في المرحلة الأولى للبرنامج عند تصميم الشاشات والقوائم المختلفة للبرنامج، وذلك لكل أداة على حدة أثناء تصميمها أو وضعها على نافذة البرنامج Form.

وتتم عملية ضبط الخصائص هذه من خلال الخطوات التالية :

١ - بالطبع نافذة البرنامج Form موجودة ونشطة الخاصة بالبرنامج الحالي.

٢ - اختيار الأداة المراد تحديد خصائصها على نافذة البرنامج من خلال نافذة الأدوات ToolBox، وذلك بتحريك مؤشر الفأرة على الأداة المطلوبة والضغط على الأداة.

٣ - التحرك بالمؤشر مرة أخرى إلى نافذة البرنامج Form لوضع الأداة في المكان الذي تختاره أو يناسب البرنامج.

٤ - رسم الأداة أو كتابة محتوياتها على نافذة البرنامج إذا لزم ذلك.

٥ - فتح نافذة الخصائص الملحقة بالبرنامج كما نعرف من خلال الضغط على F4 أو قائمة المشاهدة View.

٦ - التنقل داخل نافذة الخصائص للخاصية التي المطلوبة من خلال شريط التمرير الموجود على يمين النافذة مع مؤشر الفأرة، وبالطبع يوجد قسمين بهذه النافذة هما كما عرفنا قسم أو <u>خانة اسم الخاصية</u> (على اليسار) و قسم أو <u>خانة إدخال قيمة الخاصية</u> (على اليمين).

٧ – بعد ذلك يتم تنشيط الخاصية بظهور اللون الداكن أو الأزرق عليها، وبالتالي يمكن تحريك المؤشر على نفس سطر الخاصية إلى (خانة إدخال قيمة الخاصية) ويتم التعامل معها بإحدى الطريقتين التاليتين:

أ – الكتابة مباشرة للبيانات الجديدة من خلال لوحة المفاتيح.

ب – أو الضغط على قضيب الانزلاق الخاص بهذه الخانة لفتح قائمة منسدلة خاصة بها، واختيار البيانات المناسبة لك بدلاً من الإدخال اليدوى، (وللعلم لا يظهر قضيب الانزلاق لكل الأدوات) أى الطريقة (ب) ليست لكل الأدوات.

٨ – بعد إجراء هذا التعديل ستحتفظ الأداة النشطة بالتعديلات التى تم تغييرها فى نافذة الخصائص المرتبطة بهذه الأداة.

مثال ١ :

يعطى برنامج VB عند تنفيذ أمر إنشاء المشروع اسم افتراضي لنافذة البرنامج Form وهو Form1 على أساس انه أول نافذة بهذا البرنامج، ولكن يمكن تغيير هذا الاسم حسب البرنامج الخاص بك وكما تريد، على سبيل المثال فيمكن التغيير إلى الاسم التالى : " Main Menu ".

لتنفيذ ذلك يمكن إتباع الآتي :

١ – الذهاب إلى قائمة الخصائص والبحث عن خاصية Caption وهى الخاصة بتحديد اسم نافذة البرنامج أو الأداة المرفقة بها، والضغط عليه لتنشيطها وبالتالى سيتغير لونها.

٢ – التحرك بالمؤشر إلى الخانة المجاورة لهذه الخاصية والضغط على الزر الأيسر للفأرة لتثبيت المؤشر فى الموضع الذى تريده.

٣ – ثم الكتابة مباشرة للبيانات الجديدة التى تريد إدخالها، ثم الانتهاء من ذلك بتحريك المؤشر إلى أى مكان أخر بالنوافذ.

مثال ٢ :

هذا المثال خاص بخصائص الأدوات.

فى حالة تحديد أو تغيير خاصية أداة موجودة على نافذة البرنامج، وأيضاً عند إنشاء الخاصية على نافذة البرنامج، فإن برنامج VB يعطى هذه الأداة خصائص افتراضية من عنده لها، وبالطبع يمكن لمعد البرنامج أو المستخدم التعديل أو التغيير فى خصائص تلك الأدوات.

وذلك بنفس الأسلوب السابق مع تنشيط الأداة المراد تغيير خصائصها أولاً، ثم الانتقال إلى نافذة الخصائص وإجراء التعديل فى الخاصية المطلوب تعديلها.

على سبيل المثال :

يمكن التحكم فى إظهار الأدوات أو عدم إظهارها أثناء تشغيل البرنامج، وذلك من خلال خاصية Visible، لتنفيذ ذلك يتبع الآتى:

١ – تنشيط الأداة المطلوبة.

٢ – لذهاب لنافذة الخصائص.

٣ – لضغط على قضيب الانزلاق للبحث عن خاصية visible، وتنشيطها.

٤ – تحريك المؤشر إلى خانة الإدخال الخاصة بها (المجاورة لها) وتثبيته وسنجد القيمة الافتراضية لها True للإشارة إلى ظهور هذه الأداة أثناء تنفيذ البرنامج.

٥ – ثم الضغط على قضيب الانزلاق الموجود بهذه الخانة وظهور مربع يحتوى على True (صحيح) أو False (خطأ)، وبالتالي يمكن تحريك المؤشر للحالة المطلوبة وليكن False لعدم ظهور الأداة .

ملاحظة:

أثناء الإدخال يمكن استخدام مفاتيح الأسهم أو أحد المفاتيح التالية Del , Ins , Home , End, Caps لإجراء أى تعديلات فى خانة الإدخال.

٢ - ضبط الخصائص أثناء تشغيل البرنامج :

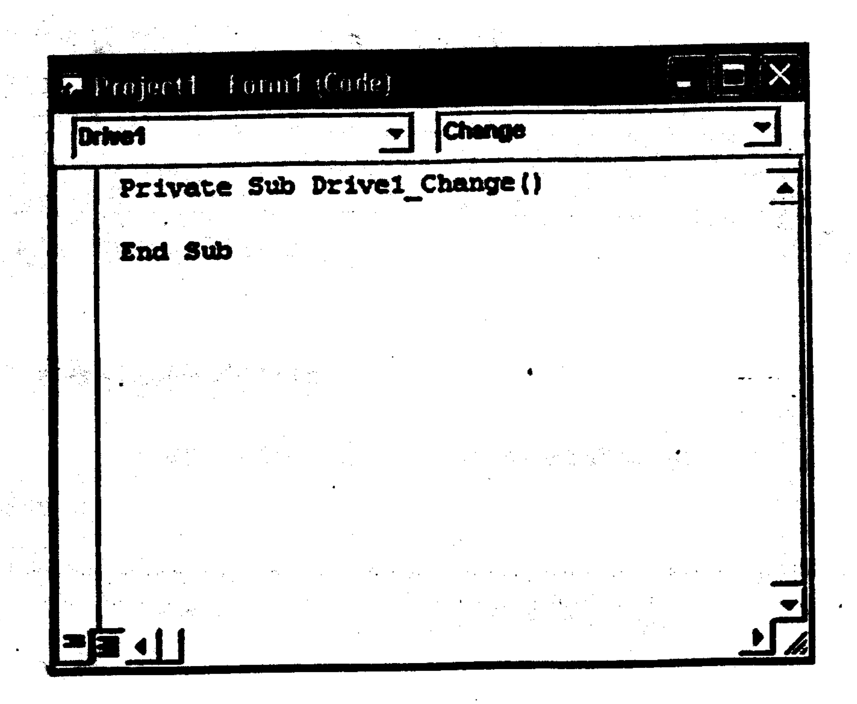
يتم تنفيذ بعض الخصائص من خلال هذه الطريقة، والسبب أن هذه الخصائص لا تنشط إلا فى طور التشغيل للبرنامج (تنفيذ أمر Run)، منها خصائص التحكم فى وحدات الإدارة Drives أو فى الأدلة Paths.

ويتم تنفيذ ذلك بكتابة أوامر لغة البيسيك المرئى فى نافذة الكود ويتم إظهار تلك النافذة بإحدى الطريقتين الآتيتين :

١ - تنشيط نافذة الكود Code من نافذة المشاهدة View.

٢ - الضغط على الزر الأيسر للفأرة مـرتين بعـد تنشـيط الأداة أو النافذة المراد إعداد البرنامج (الأوامر) لها.

وعندئذ ستظهر نافذة بالشكل التالى:



مثال :

يمكن كتابة الأمر التالى لتنشيط وحدات الإدارة الموجودة على الجهاز عند تشغيل (تنفيذ) البرنامج.

```
Drive1.Left = 300
```

بين الأمرين التاليين :

```
Private Sub Drive1_Change()

End Sub
```

- ونتيجة لتنفيذ هذا الأمر ستنشط أداة وحدة إدارة الوحدات (الأقراص)، ويمكن الضغط على السهم الموجود على شريط التمرير الموجود على يسار الأداة لاستعراض كل الوحدات المتوفرة على الجهاز، وبالتالى تحريك مؤشر الفأرة عليها واختيار الوحدة التى تريد التعامل معها.

بعد انتهاء التعديل للخصائص أثناء مرحلة تنفيذ البرنامج لأن هذا التعديل لا يؤثر على الأدوات أثناء مرحلة التصميم للأدوات والبرنامج فى البداية ولذا إذا أوقفت البرنامج وأعدت تشغيله من جديد أو عدت إلى مرحلة التصميم فستجد النافذة والأدوات التى قمت بتغييرها قد عادت إلى ما كانت عليه من قبل، (راجع قواعد كتابة صيغة الخصائص)

٤ - ٥ قاعدة (صيغة) الخصائص

```
Controlname.PropertyName = New Value
```



شرح القاعدة :

Controlname : تشير إلى اسم الأداة التى ترغب فى تغييرها (الأدوات الموجودة فى صندوق الأدوات Tools Box).

. : تشير إلى النقطة العادية (فاصل بين الأداة والخاصية المـراد تعديلها).

PropertyName : تشير إلى اسم الخاصية التى ترغـب فـى تعـديل قيمتهـا (الخصائص الموجودة فى نافذة الخصائص (Properties).

= : تشير إلى علامة التساوي للقيم.

New Value : تشير إلى القيمة الجديدة للخاصية والتى يتم إدخالها فى نافـذة الخصائص (Properties) وبالتحديد فى (خانة الإدخال)، حيث يتم كتابة قيمتها أو الحصول عليها من قوائم أخرى خاصة بالأداة.

✍ ملاحظة :

- تنفيذ هذه القاعدة يتم بطريقة آليه فى مرحلة التصميم وذلك باختيارك للأداة ثم اختيار خصائصها، ولكن يمكن تنفيذها يـدوياً مـن خـلال المستخدم عن طريق قيامك بكتابة صيغة الأداة بنفسك فـى مرحلـة التنفيذ للبرنامج كما عرفنا من قبل.
- توجد بعض الخصائص الثابتة أو المشتركة التى توجد فـى معظـم الأدوات مثل الخصائص الخاصة بالموقع والخط والألوان بالإضـافة إلى وجود بعض الخصائص المرتبطة أو الخاصة بـبعض الأدوات أو خاصة بأداة معينة (أي لا توجد فى بقية الأدوات) مثل خاصـية وحـدات الإدارة Drive أو خاصـية الأدلـة Path أو خاصـية النص Text.

٤ – ٦ شرح بعض الخصائص:

١ - خاصية الاسم : Name

الغرض منه :

اختيار أو تحديد اسم الأداة أثناء تصميم أو كتابة البرنامج وذلك للتعامل معها من خلاله، وهذه الخاصية عامة الاستخدام لكل الأدوات أى يجب أن يكون لكل أداة اسم معين يظهر هذا الاسم على الأداة عند تصميمها.

ويعطى أو يمنح اسم أو عنوان(النافذة أو الأداة) افتراضيا من النظام أو قد تغييره أنت أثناء التصميم.

مثال :

نافذة البرنامج Form تأخذ اسم افتراضى Form1 للنافذة أو الشاشة الأولى فى المشروع وفى حالة تكرار الإنشاء لنافذة أخرى سيعطى النظام اسم جديد لنافذة برنامج جديدة وهو Form2 وهكذا بالنسبة للنوافذ. أما بالنسبة للأدوات فمنها على سبيل المثال أداة الأمر Command وسوف يعطى اسم لهذه الخاصية وهو Command1 وفى حالة تكرار الخاصية على نفس النافذة سيعطى Command2 وهكذا.

ويستحسن عند تنفيذ ذلك أن تعطى اسما يشير إلى معنى للأداة أو النافذة المستخدمة.

هذه الخاصية موجودة أثناء : مرحلة تصميم البرنامج

٢ - خاصية العنوان: Caption

الغرض منها :

تحديد العنوان الذى سيظهر فوق الأداة وتحدد أيضاً العنوان الذى سيظهر فوق النافذة وبالتحديد فوق شريط العنوان Title Bar.

هذه الخاصية متوفرة للأدوات التالية : أداة العنـــوان ، أداة قاعـــدة البيانـــات، زر الأوامر، زر الخيار، خانة الاختيار، النافذة Form.

مثال :

يمكن التحكم فى هذا الاسم من خلال قائمة الخصائص أو من خـــلال أوامـــر البرنامج بكتابة الأمر التالى :

```
Label1.Caption = "edit "
```

كما نعرف Label1 تعنـــى الأداة النصـــية (A) ، Caption تعنـــى اســـم الخاصية، edit تشير للقيم الجديدة لهذه الأداة.

هذه الخاصية موجودة أثناء : مرحلتي التصميم والتنفيذ.

٣ - خاصية الألوان :

خاصية لون الخلفية BackColor

خاصية لون الكتابة الأمامية ForeColor

الغرض منها :

تحديد اللون المناسب الذى تختاره للشاشات أو القوائم أو محتوياتهما الخاصية ببرنامجك.

مثال :

```
Private Sub Label1_Click()
    Label1.Caption   = " edit "
    Label1.BackColor       = QBColor(2)
    Label1.ForeColor       = QBColor(5)
End Sub
```

حيث تشير الأرقام الموجودة بجوار الخصائص إلى رقم اللون كما يلى:

رقم ١ يشير إلى اللون الأزرق الداكن
رقم ٢ يشير إلى اللون الأخضر
رقم ٣ يشير إلى اللون الأزرق
رقم ٤ يشير إلى اللون الأحمر
رقم ٥ يشير إلى اللون البنفسجى

هذه الخاصية موجودة أثناء : مرحلتي التصميم والتنفيذ.

٤ - خاصية الإخفاء Visible

الغرض منها :

إخفاء بعض الأدوات التى تظهر على نافذة برنامجك (حسب حاجة معد البرنامج أو المستخدم)، وتستخدم مع أغلب الأدوات المستخدمة على نافذة البرنامج.

مثال :

```
Private Sub Label1_Click()
    Label1.Visible = False
End Sub
```

ويشير هذا المثال إلى إلغاء أو عدم ظهور أداة العنوان label1 أثناء تنفيذ البرنامج، وفى حالة كتابة True فستظهر هذه الأداة.

هذه الخاصية موجودة أثناء : مرحلتي التصميم والتنفيذ.

٥ - خصائص الخطوط

خاصية نوع الخط	FontName
خاصية حجم الخط	FontSize
خاصية خط أسفل الكلمة	FontUnderLine
خاصية الخط السميك	FontBold
خاصية الخط المائل	FontItalic

الغرض منها :

بالطبع التحكم فى درجة ونوع الخطوط كما يظهر من أسماء الخصائص السابقة هذه الخاصية موجودة أثناء : مرحلتي التصميم والتنفيذ.

تستخدم هذه الخصائص فى النوافذ والأدوات التى لها علاقة بظهور الخطـــوط مثل نافذة البرنامج Form أو أداة العنوان أو النص أو الأوامر أو خانة الاختيار.

ملاحظة:

تستخدم مع خاصية حجم الخط أرقام تحددها من النافذة التى ستظهر بجوار الخاصية فى خانة الإدخال.

تستخدم مع خاصية اسم الخط أسماء للخطوط تحددها من النافذة التى ستظهر بجوار الخاصية فى خانة الإدخال.

أما باقى الخصائص الأخرى الخاصة بالخطوط فستظهر فـــى خانـــة الإدخال لها إحدى الكلمتين True أو False (صحيح أو خطأ) وذلـــك لتنشيط أو عدم التنشيط للأداة أى تريد أو لا تريد هذه الخاصية فى الخط.

٦ - خاصية الأسطر المتعددة MultiLine

الغرض منها :

إتاحة أو تعدد الأسطر فى خانة الأداة للتمكن من الكتابة بها أكثر من سطر.

مثال :

فى حالة اختيار أداة (ab) صندوق النصـوص ووضـع الأداة علـى نافـذة المشروع، ثم الذهاب لهذه الخاصية فى نافذة الخصائص، سنجد خيارين هما صـح أو خطأ True أو False، وبالتالى يمكن اختيار True.

وترتبط هذه الخاصية بخاصية النص Text للأداة النصية حيث يـتم الضـغط عليها وهنا سيظهر صندوق أعلى خانة الإدخال لهذه الخاصية يمكن الكتابـة فيـه لعدد من الأسطر، وبالتالى تتيح هذه الخاصية تعدد الأسطر فى الأداة التى يتم فيها الكتابة لنصوص.

هذه الخاصية موجودة أثناء : مرحلتي التصميم والتنفيذ.

٧ - خاصية الموقع على النموذج :

Top

Left

الغرض منها :

التحكم فى موقع الأدوات على نافذة البرنامج من حيث :

Top تشير إلى تحريك نافذة البرنامج إلى الطرف العلوى على الشاشة.

Left تشير إلى تحريك نافذة البرنامج إلى الطرف الأيسر على الشاشة.

مثال :

```
Form2.top=0
Form2.left=0
```

هذه الخصائص ستحرك النافذة إلى أعلى الشاشة من اليسار (على السطر الأول بها).

مع العلم بأن الوضع الافتراضي لها يأخذ قيمة فى حدود ١٥٥٠ لخاصية Top وذلك لوضع النافذة أسفل شريط الأدوات الخاص بالبرنامج، وبالطبع يمكن التحكم فى هذه القيمة بكتابتها فى خانة الإدخال بالزيادة للقيمة سوف تنزل النافذة إلى أسفل الشاشة.

وأيضاً خاصية Left ستحرك النافذة إلى جهة اليسار من الشاشة (القيمة الصغرى لأعلى اليسار).

هذه الخاصية متوفرة أثناء : مرحلتي التصميم والتنفيذ.

٨ - خاصية الحجم :

Height

Width

الغرض منها :

التحكم فى ارتفاع وحجم الأداة أو نافذة البرنامج على الشاشة، حيث خاصية Height خاصة بارتفاع الأداة، وخاصية Width خاصة بحجم الأداة.

مثال :

فى حالة كتابة قيمة ١٥٥٠ لخاصية Height الارتفاع لنافذة البرنامج (بعد تنشيطها) سنجد أن الارتفاع لها سيقل إلى تقريباً ثلث النافذة، وفى حالة الزيادة للقيمة سيزداد الارتفاع، والوضع الطبيعي للنافذة فى حدود ٤٣٠٠ وبالطبع يمكن أن تزيد حتى تصل إلى نهاية الشاشة من أسفل.

بالنسبة لخاصية Width والخاصة بعرض النافـذة أو الأداة، بـالطبع تنشـيط النافذة أو الأداة المراد تغيير خاصية العرض لها، ثم كتابة الـرقم الـذى تريـده، وفى حالة إدخال قيمة أقل من القيمة الحاليـة سـتجد أن النافـذة أو الأداة سـيقل عرضها إلى جهة اليسار.

مثال أخر :

للتعديل أثناء مرحلة التنفيـذ للبرنـامج، مـن خـلال نافـذة الكـود Code، وهنا يتم تنشيط أداة عرض أدلة وحدة الإدارة DriveListBox المراد التحكم فـى حجمها، والنقر المزدوج عليها لفتح نافذة الكود ثم كتابة الكود المطلـوب وتحديـد الحجم الذى يرغب فيه المصمم :

```
Private Sub Dir1_Change()
    Dir1.Width = Dir1.Width * 2
End Sub
```

هذه الخاصية متوفرة أثناء : مرحلتي التصميم والتنفيذ.

٩ - خاصية النص Text :

الغرض منها :

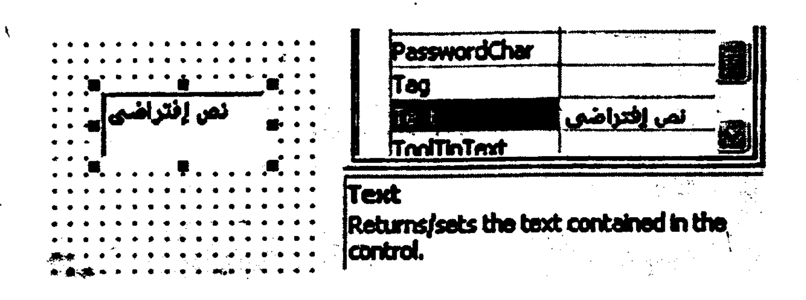
إتاحة كتابة بعض الكلمات داخل الأداة النصية، تظهر بشكل افتراضـى فـور ظهور الأداة.

مثال :

الذهاب للخاصية Text فـــى نافـــذة الخصــــائص وتنشـــيط خانـــة الإدخـــال، وهنا سيظهر صندوق أعلى خانة الإدخال لهذه الخاصية ويمكن الكتابة فيه للـــنص الذى تريد ظهوره على نافذة البرنامج للخاصية النشطة.

هذه الخاصية موجودة أثناء : مرحلتي التصميم والتنفيذ.

وفيما يلى شكل لها بعد كتابة النصـــوص التـــي تريـــدها داخـــل الصـــندوق الخاص بها:



١٠ – خاصية شكل المؤشر Mouse Pointer :

الغرض منها :

التحكم فى شكل مؤشر الفأرة على النافذة، حيث يمكن تغيير شكل المؤشر بـــين عدد ١٥ شكل أثناء تنفيذ البرنامج.

مثال:

يمكن التحرك للخاصية فى نافذة الخصائص والانتقال إلى خانة الإدخال والنقر على شريط التمرير الخاص بها، عندئذ سيظهر مجموعة من الخيارات تبـدأ مـن الوضع الافتراضي للمؤشر ثم رقم ١ إلى ١٥ ويوجد رقم ٩٩ وذلك لإتاحة إضافة شكل للمؤشر غير متوفر بالأشكال الافتراضية.

- وبالطبع رقم ١ يشير إلى شكل المؤشر العادى (السهم).
- ورقم ٢ يشير للشكل (+).
- ورقم ٣ يشير إلى الشكل (I) ...الخ. راجع ذلك بالتنفيذ العملى.

هذه الخاصية موجودة أثناء : مرحلتي التصميم والتنفيذ.

١١ - خاصية التمكين Enabled :

الغرض منها :

عدم تمكين المستخدم من التعامل مـع الأداة التـى خاصـيتها غيـر ممكنـة للاستخدام False.

حيث ستظهر هذه الخاصية بلون باهت على نافذة البرنـامج، معنـى ذلـك أن المستخدم لا يستطيع استخدامها (الأداة) إلا إذا كانت خاصية Enabled ممكنـة الاستخدام أى True.

مثال :

```
Private Sub Text1_Change()
    Text1.Enabled = " False"
End Sub
```

أو يتم ذلـــك بـــالطبع مـــن خـــلال نافـــذة الخصـــائص باختيـــار الخاصـــية واختيار False بدلاً من True.

هذه الخاصية موجودة أثناء : مرحلتي التصميم والتنفيذ.

١٢ – بعض الخصائص الموجودة أثناء مرحلة التنفيذ :

أ – خاصية وحدات الإدارة Drive :

الغرض منها :

عرض وحدات الإدارة الموجودة على الجهاز داخل قائمة منسدلة :

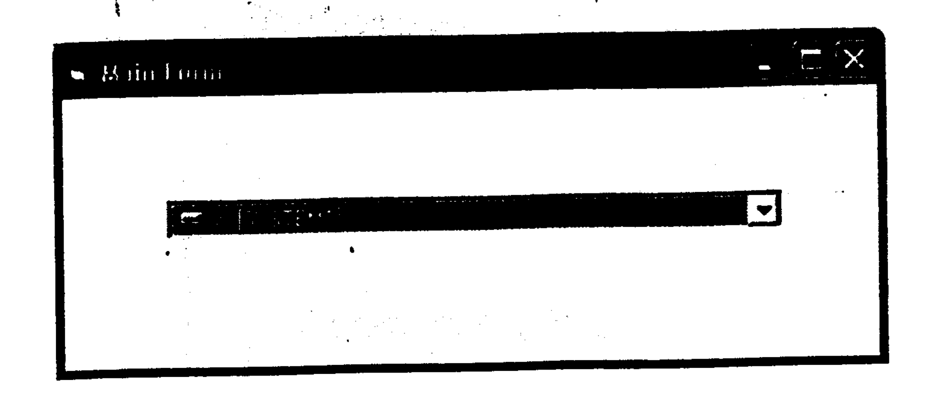
مثال :

```
Private Sub Drive2_Change()
    Drive2.Wedith = 5000
End Sub
```

هذه الخاصية موجودة أثناء : مرحلة التنفيذ.

ناتج التنفيذ للأمر السابق :

عرض جميع الوحدات الملحقة بالجهاز فى صندوق خاص بها مع تحديد عرض الصندوق كما سيتضح فى الشكل التالى:



ب - خاصية الأدلة Path (Directory) :

الغرض منها :

عرض الأدلة الموجودة على وحدة الإدارة النشطة.

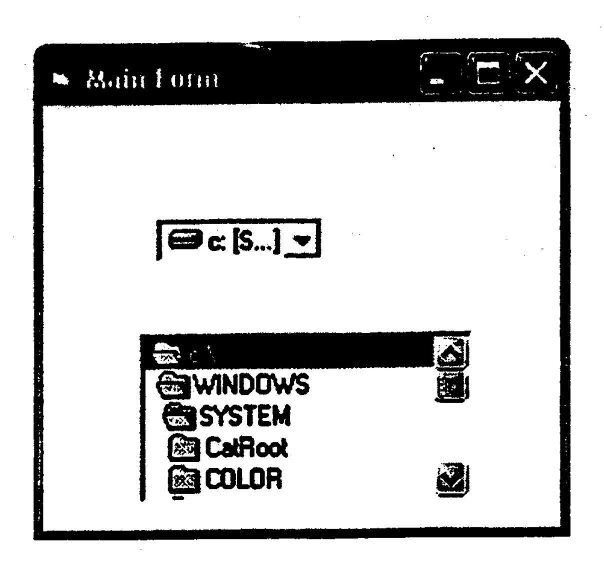
مثال :

```
Private Sub Dir1_Change()
        Dir1.Width = Dir1.Width * 2
End Sub
```

هذه الخاصية موجودة أثناء : مرحلة التنفيذ.

ناتج التنفيذ للأمر السابق :

عرض أسماء الأدلة فى الصندوق الخاص بها مضاعف الحجم كما سيتضح فى الشكل التالى:



ويمكن إضافة الكود التالى فى الأداة DriveListBox حتى يتم ربطها برمجيا بالأداة DirListBox، وذلك ليتم عرض الأدلة الموجودة بالوحدة التى يقوم المستخدم باختيارها من DriveListBox :

```
Private Sub Drive1_Change()
    Dir1.Path = Drive1.Drive
End Sub
```

ج - خاصية عرض الملفات File :

الغرض منها : عرض الملفات الموجودة على الدليل النشط الحالى.

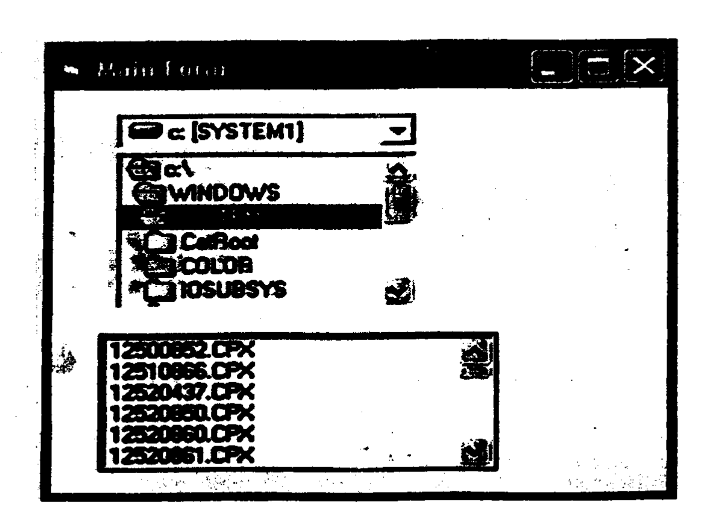
```
Private Sub File1_Click()
    File1.Path="c:\"
End Sub
```

ناتج التنفيذ للأمر السابق :

عرض أسماء الملفات الموجود على الوحدة C فى الصندوق الخاص بها.

ويمكن إضافة الكود التالى فى الأداة DirListBox حتى يتم ربطهـا برمجيـا بالأداة FileListBox، وذلك ليتم عرض الملفات الموجودة بالدليل الـذى يقـوم المستخدم باختياره من DirListBox :

```
Private Sub Dir1_Change()
    File1.Path = Dir1.Path
End Sub
```



الفصل الخامس

باقي النوافذ

يتناول هذا الفصل المحتويات التالية::

- نافذة المشروع Project
- نافذة الكود Code
- نافذة موضع النموذج Layout
- نافذة الألوان Color Palette

٥ - ١ نافذة المشروع Project

الغرض منها :

عرض قائمة بأسماء النماذج (نوافذ البرنامج Forms) والمشاريع والبرامج الرئيسية أو الفرعية (Modules) وذلك حتى يتمكن المصمم من التنقل وفتح وحذف وإضافة وحفظ وطباعة هذه النماذج، ويتم عرض النماذج داخل قائمة نافذة المشروع بحسب أسبقية إنشاؤها.

*** كيفية إظهارها :**

الطريقة الأولى :

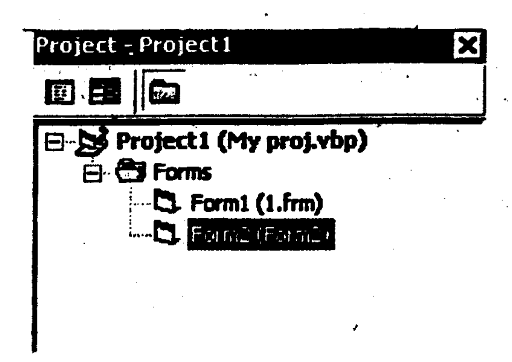
تظهر هذه النافذة افتراضيا (أى تمنح مباشرة من برنامج VB عند إنشاء مشروع جديد).

الطريقة الثانية :

من خلال النافذة الفرعية View :

وذلك فى حالة إغلاق النافذة من قبل المستخدم أثناء التشغيل للبرنامج، ويمكن فتحها من جديد من خلال نافذة المشاهدة View الموجودة فى القائمة الرئيسية لبرنامج VB، وذلك بالتحرك بالمؤشر إلى أمر Project Explorer بها والضغط عليه مره واحدة، عندئذ ستظهر النافذة على الشاشة.

وفيما يلى شكل هذه النافذة :



* **مكونات النافذة :**

تتكون النافذة من الأجزاء التالية :

أولاً : سطر العنوان :

يحتوى هذا السطر على كلمة Project – Project1 والذى يدل على محتويات المشروع الحالى.

بالإضافة إلى زر X والذى يمكن إغلاق النافذة من خلاله.

ثانياً : أزرار المشاهدة لنوافذ :

(نافذة البرنامج View Object، نافذة الكـود View Code ، عرض المجلدات Toggle Folder)

يحتوى هذا الجزء على ثلاث أزرار يمكن النقر عليهم لتشغيل الآتي:

أ - مفتاح نافذة البرنامج View Object:

من خلال هذا المفتاح يمكن عرض نافذة البرنامج النشطة على الشاشة لمشاهدة محتوياتها والتأكد منها وذلك فى حالة الحاجة للتعديل فيها بأى شكل.

للتنفيذ يتبع الآتي :

- فتح المشروع المراد التعامل معه.
- ستظهر جميع نوافذ البرنامج Forms ذات النوعية، FRM في صندوق أو قائمة محتويات نافذة المشروع.
- وبالتالى يجب تنشيط نافذة البرنامج المراد التعامل معها أو عرضها، وذلك بتحريك المؤشر على نافذة البرنامج التى تريدها.
- ثم يتم النقر على مفتاح أو زر View Objects، عندئذ ستفتح نافذة البرنامج المختارة على الشاشة ويمكنك التعامل معها كما تريد، وهكذا بالنسبة للنوافذ الأخرى، مع ملاحظة أن النقر المزدوج على اسم النموذج يؤدى نفس الغرض الخاص بهذا الزر.

✍ ملاحظة :

قد تحتوى قائمة أو صندوق النافذة على ملفات أخرى ذات امتداد أخر لا يمكن فتحها بواسطة هذه النافذة، لأنه كما قلنا من قبل تتعامل View Objects مع نوافذ البرنامج فقط ذات الامتداد FRM.

ب - مفتاح نافذة الكود View Code :

الغرض منها :

فتح نافذة الكود الخاصة بكتابة الأوامر والتعليمات المختلفة الخاصة بنافذة البرنامج أو ببرنامج المشروع الرئيسى أو البرامج الفرعية، وهذه النافذة خاصة بالنموذج المحدد حالياً بالنافذة.

أو أيضاً يمكن عرض نافذة الكود الخاصة بالبرنامج النشط على الشاشة لمشاهدة محتوياتها من الأوامر المختلفة السابق كتابتها من قبل، وكذلك أيضاً في حالة الحاجة للتعديل فيها بأى شكل.

* للتنفيذ يتبع الآتي :

- فتح المشروع المراد التعامل معه.
- ستظهر جميع نوافذ البرنامج Forms ذات النوعية FRM فى صندوق أو قائمة محتويات نافذة المشروع.
- وبالتالى يجب تنشيط نافذة البرنامج المراد التعامل معها أو عرضها، وذلك بتحريك المؤشر علي نافذة البرنامج التى تريدها.
- ثم يتم النقر على مفتاح أو زر View Form، عندئذ ستفتح نافذة البرنامج المختارة على الشاشة ويمكنك التعامل معها كما تريد، وهكذا بالنسبة للنوافذ الأخرى.
- عندئذ يمكن النقر على مفتاح View Code وسيظهر نافذة الكود الخاصة بنافذة البرنامج الحالى أو برنامج اللغة الجارى التعامل معه.
- وبالتالى ستظهر جميع التعليمات أو الأوامر الخاصة بها، ويتاح لك المشاهدة أو التعديل كما تريد فى محتويات البرنامج.

ملاحظة :

يمكن إنشاء أو فتح أى عدد من نوافذ البرنامج Form أو برامج اللغة Modules في المشروع الواحد أثناء التصميم للبرنامج، ولكن لا يمكن فتح أكثر من مشروع واحد فى كل مرة.

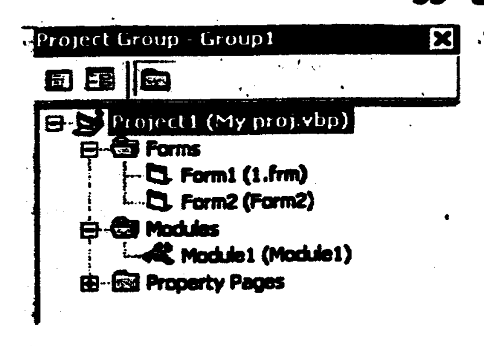
ب - مفتاح عرض المجلدات Toggle Folder:

الغرض منها :

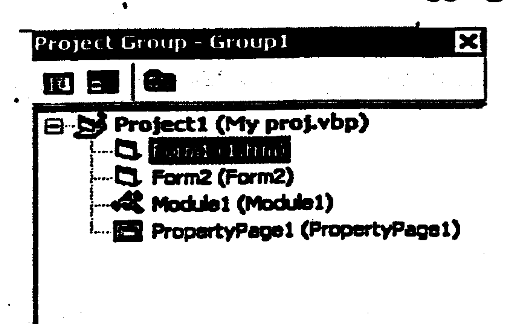
عرض النماذج أو البرامج الفرعية داخل مجلدات خاصة بها يمكن ضمها معاً أو إعادة سردها داخل تلك القائمة، مما يساعد المصمم على سرعة التحكم والوصول لعناصر برنامجه المختلفة.

والشكل التالى يوضح تأثير الزر قبل وبعد الضغط عليه :

قبل الضغط على الزر :



بعد الضغط على الزر :



ثالثاً : قائمة (صندوق) المحتويات :

تحتوى هذه القائمة على أسماء نوافذ البرنامج والبـرامج والملفـات المختلفـة الأنواع التى تلحق أو تنشأ بالمشروع أو التخطيط الجارى التعامل معه، والتى يـتم عرضها على شكل شجرة وأفرع.

وبالطبع يحدد اسم الملف ونوعه كل على سطر مستقل داخل النافذة.

ملاحظة :

قد يظهر شريط تمرير رأسي على هذه النافذة فى حالة وجود ملفات أو نوافذ كثيرة بالمشروع المفتوح، وبالتالى يمكن للمسـتخدم اسـتخدامه لعرض أسماؤها واختياره ما يريد منها.

٥ - ٢ نافذة الكود (برنامج الأحداث) Code

الغرض منها :

كتابة التعليمات (الأوامر) الخاصة بلغة البيسيك المرئى والمرتبطة بحدث معين يتم تنفيذه بواسطة المستخدم.

وهذا الحدث قد يتمثل فى مثلاً : قيام المستخدم بالنقر أو النقر المزدوج من خلال الفأرة أو الضغط من خلال لوحة المفاتيح.

وبالطبع هذا الحدث يرتبط أثناء تنفيذه ببرنامج معين خاص بأداة معينة أو نافذة معينة.

ويكتب التعليمات المرتبطة بالحدث بين جملتي برنامج فرعى:

```
Sub ....
End Sub
```

ويتكون الحدث من الآتي :

١ - اسم الأداة المرتبطة بالحدث أو التى وقع عليها الحدث وتأتى بعد كلمة Sub مثل كلمة File1.

٢ - الحدث نفسه مثل الضغط ()Click.

٣ - مجموعة الأوامر المراد تنفيذها إذا وقع هذا الحدث.

كيفية إظهارها :

يمكن أن تظهر نافذة خاصة بها من خلال الطرق التالية :

١ – النقر المزدوج للزر الأيسر للفأرة:

– حيث يمكن أولاً تنشيط الأداة أو النافذة المراد كتابة البرنامج لها أو التعليمات.

– ثم التحريك المؤشر على سطر العنوان الخاص بنافذة المشروع إلى كلمة Code.

– ثم النقر عليها، عندئذ ستفتح شاشة الكود أو نافذة الكود ليتم كتابة الأوامر المرتبطة بالأداة النشطة.

٢ – القائمة الفرعية View.

– حيث يتم تنشيط الأداة المراد كتابة تعليمات أو أوامر معينة لها أولاً.

– ثم تحريك المؤشر على قائمة المشاهدة View وفتحها.

– ثم تحريك المؤشر إلى أمر Code والضغط عليه.

–عندئذ ستفتح نافذة أو شاشة الكود ويمكن كتابة الأوامر بين سطرى الحدث المراد تنفيذه.

٣ – من خلال نافذة المشروع Project Explorer :

– حيث يتم أولاً اختيار نافذة البرنامج المراد التعامل معها من ضمن قائمة هذه النافذة لأنها يمكن أن تحتوى على أكثر من نافذة برنامج، وذلك بتحريك المؤشر عليها والضغط عليها لفتحها.

– ثم التحرك بالمؤشر إلى زر View Code الموجود فى أعلى نافذة المشروع والنقر عليه.

– عندئذ ستظهر نافذة الكود على الشاشة مكان نافذة البرنامج المفتوح من قبل.

٤ – من خلال لوحة المفاتيح

– أيضا بعد التنشيط لنافذة البرنامج المراد التعامل معها.

– يمكن الضغط مباشرة على مفتاح F7.

– عندئذ ستفتح نافذة الكود ويمكن التعامل معها.

** خطوات التنفيذ العملى لها :

– تنشيط نافذة البرنامج Form.

– تنشيط النافذة أو الأداة المراد كتابة تعليمات لها لتنفذ أثناء تنفيذ البرنامج (أى تحريك مؤشر الفأرة عليها).

– النقر أو الضغط مرتين متتاليتين عليها.

– عندئذ ستظهر نافذة الكود Code window، وبها سطرين للكتابة بينهما وهما :

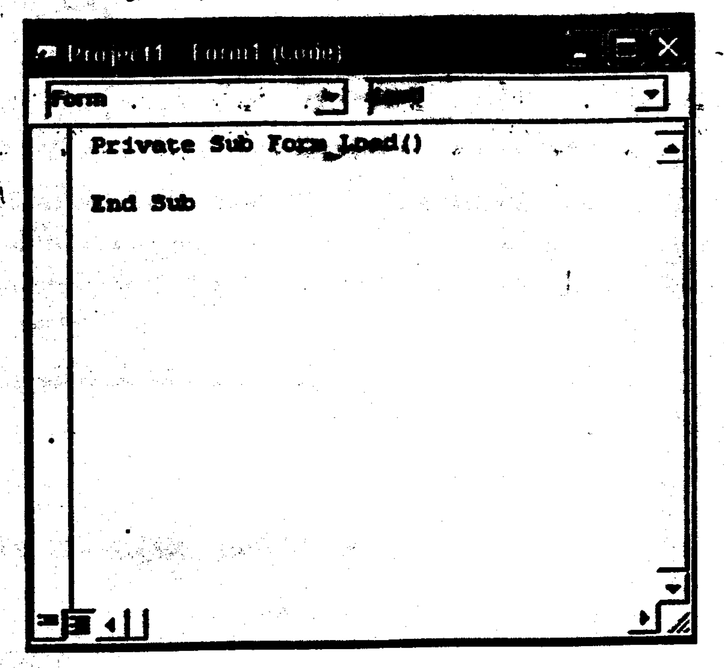
```
Private Sub Form_Load()

End Sub
```

وبالطبع يتغير Form_Load تبعاً لنوع الأداة والحدث المراد إضافة برنامج له.

– وبالتالى تحريك المؤشر بين السطرين السابقين لبدء كتابة الكود المطلوب.

* وفيما يلى شكل هذه النافذة عند النقر المزدوج على جزء فارغ بالنموذج:



* مكونات النافذة :

تتكون النافذة من الآتي :

١ - السطر الأول (سطر العنوان Title Bar) :

يحتوى هذا السطر على عنوان نافذة البرنامج التى يتم التعامل معها ويوجد فى الجانب الأيسر، وهو يمكن تغيره من نافذة الخصائص، وفى الجانب الأيمن العلوى توجد علامات التعامل العادية الخاصة ببرنامج النوافذ مثل علامة X للغلق وعلامة التكبير (المربع) وعلامة التصغير للنوافذ على سطر المهام (-).

٢ – السطر الثانى :

هذا السطر خاص بعرض قائمة منسدلة بأسماء الأدوات التى يتم إضافتها على نافذة البرنامج وقائمة منسدلة أخرى بالأحداث المرتبطة بالأداة المحددة حاليا، حيث يتم ربط ذلك ببرامج فرعية مرتبطة بالأدوات والأحداث الخاصة بها.

حيث تستخدم هذه النافذة لإدخال الأوامر والتعليمات التى تريد تنفيـذها عنـد وقوع الأحداث التى تصدرها كما عرفنا من خلال النقر أو النقر المزدوج لمؤشـر الفأرة أو استخدام لوحة المفاتيح أو استخدام أداة الوقت الخاصة بنافـذة صـندوق الأدوات ... الخ.

ينقسم هذا السطر إلى قسمين هما :

القسم الأول : خانة الأدوات Object

الغرض منها :

عرض أسماء الأدوات المختلفة السابق إنشائها على نافذة البرنامج (النمـوذج) فيها النافذة نفسـها، حيـث يمكـن اختيـار اسـم الأداة مـن القائمـة المنسـدلة وذلك بالتحرك على اسم الأداة من خلال مؤشر الفأرة والنقر عليه.

عندئذ سينتقل المؤشر بشكل مختلف (I) إلى جزء البرنامج الفرعى المسئول عن تشغيل هـذه الأداة (شاشـة الكـود) المرتبطـة بالحـدث المـراد تنفيـذه، وذلك لكتابة التعليمات الخاصة بتشـغيل هـذه الأداة أو لأداء أى تعـديلات فيـه (جزء الأوامر) إذا سبق كتابة هذه الأوامر.

والشكل يوضح مثال على قائمة الأدوات:

نوافذ Project و Code وـ Color

القسم الثانى : القائمة المنسدلة للأحداث (Events)

الغرض منها :

أولاً تعطى محتوياتها افتراضياً من برنامج اللغة، حيث يتم عرض أسماء جميع الأحداث المتاحة للأداة النشطة من خلال قائمة تنسدل فى حالة الضغط على السهم المتجه إلى أسفل بهذه الخانة.

حيث قد تختلف هذه الأحداث من أداة لأخرى (أى قد تختلف محتويات القائمة عند تشغيل كل أداة)، هذا بالإضافة إلي أن هذه الخانة قد تستخدم فى تغيير الحدث الذى ترغب فى الاستجابة له.

للتنفيذ :

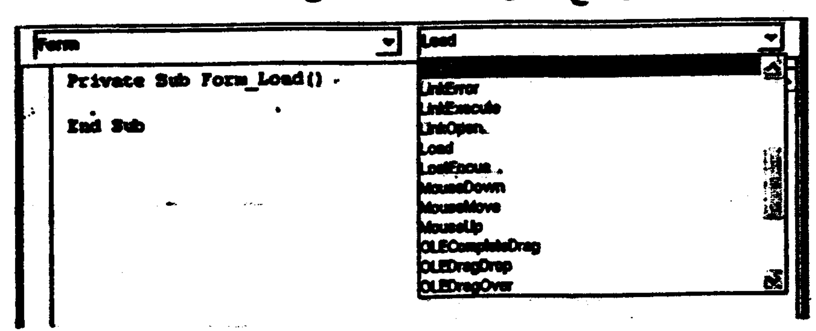
- تنشيط نافذة البرنامج Form.
- تنشيط الأداة المراد كتابة تعليمات لها لتنفذ أثناء تنفيذ البرنامج (أى تحريك مؤشر الفأرة عليها).
- النقر أو الضغط مرتين متتاليتين عليها.

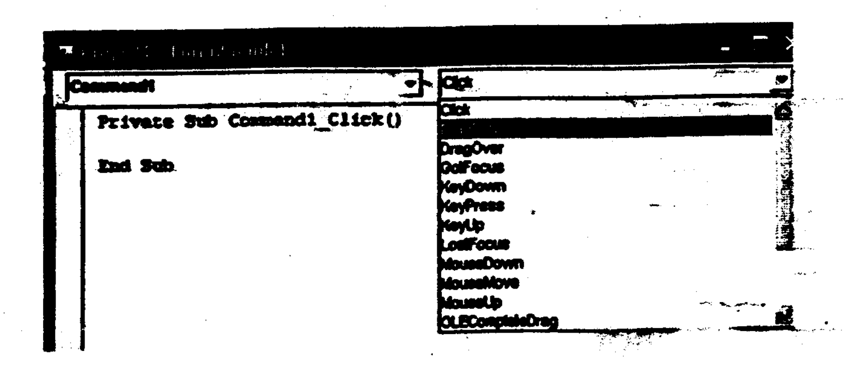
- عندئذ ستظهر نافذة الكود Code window بما تحتويه من الأقسام التاليـــة السابق معرفتها.
- تحريك المؤشر على القائمة المنسدلة والضغط على السهم المتجـه لأسـفل بيمينها.
- عندئذ ستظهر قائمة منسدلة بها الأحداث أو الإجراءات المخصصـــة لهـذه الأداة أو التى يمكن تنفيذها مع هذه الأداة.
- وبالتالى يمكن التحرك بالمؤشر على الحدث المراد تنفيذه والنقر عليه مـرة واحدة بزر الفأرة الأيسر، وهنا سيتغير اسم الحدث المجـاور لاسـم الأداة المراد كتابة برنامج فرعى لها فى سطر Sub.

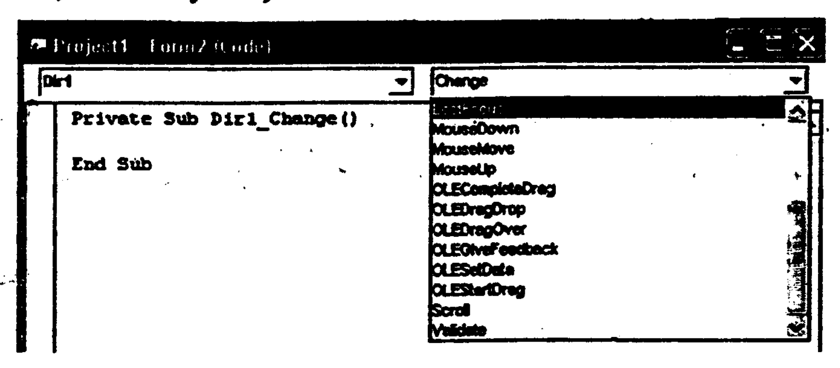
••• وفيما يلي أهم الأحداث الموجودة في هذا البرنامج

مع ملاحظة أن هذه الأحداث قد تتغير حسب الأداة أو النافذة الجاري التعامل معها، وأن تنفيذ هذه الأحداث لا يظهر إلا عند تنفيذ البرنامج والنقر أو الضغط على الأداة المخصص لها الحدث.

والأشكال التالية توضح بعض الأحداث لعدد من الكائنات:







* **تصنيف الأحداث :**

أولاً: أحداث وحدات الإدارة والأدلة وأيضا مربع النص والــ Combo :

١ – حدث التغيير Change

يتم تنفيذ أو وقوع هذا الحدث عندما تتغير الأسطوانة الحالية في قائمة وحدة الإدارة، سواء كان ذلك من خلال البرنامج أو باختيار المستخدم.

ثانياً : أحداث التعامل مع الفأرة :

٢ – حدث النقر Click

يتم تنفيذ هذا الحدث عند قيام المستخدم بالنقر أو الضغط على الأداة المخصصة لهذا الحدث مرة واحدة وبالطبع عند تنفيذ البرنامج.

٣ – حدث النقر المزدوج Dblclick

يتم تنفيذ هذا الحدث عند قيام المستخدم بالنقر المزدوج على الأداة أو النافذة المخصصة لهذا الحدث مرتين متتاليتين.

٤ - حدث سحب وإلقاء DragDrop

يتم تنفيذ هذا الحدث عند قيام المستخدم بالضغط على زر الفارة الأيسر لأداة معينة مع السحب بدون تحرير (ترك) الزر، ثم يقوم المستخدم بإلقاء الأداة في نهاية عمليه السحب في موضع معين.

٥ - حدث سحب DragOver

يتم تنفيذ هذا الحدث عند قيام المستخدم بالضغط على زر الفارة الأيسر لأداة معينة مع السحب بدون تحرير (ترك) الزر ، ويعبر بها فوق أداة أثناء عمليه السحب.

٦ - حدث حركة الفارة MouseMove

يتم تنفيذ هذا الحدث عند قيام المستخدم بتحريك الفارة فوق النافذة أو الأداة المخصصة لهذا الحدث.

٧ - حدث حركة الفارة لأسفل MouseDown

يتم تنفيذ هذا الحدث عند قيام المستخدم بالنقر على زر الفارة وقبل أن يحرره (أى أثناء عملية النقر).

٨ - حدث حركة الفارة لأعلى MouseUp

يتم تنفيذ هذا الحدث عند قيام المستخدم بالنقر على زر الفارة وتحريره (أى بعد عملية النقر).

ثالثاً : الناتجة عن استخدام لوحة المفاتيح :

٩ - حدث ضغط حرف KeyPress

يتم تنفيذ هذا الحدث عندما يتم الضغط على شفرة الأسكى ASCII من خلال لوحة المفاتيح.

١٠ - حدث المفتاح أسفل KeyDown

تم تنفيذ هذا الحدث عندما يتم الضغط على أى حرف على لوحة المفاتيح وقبل أن يتركه (يحرره) المستخدم، أى أثناء عملية الضغط.

١١ - حدث المفتاح أعلى KeyUp

يتم تنفيذ هذا الحدث عندما يتم الضغط على أى حرف على لوحة المفاتيح وبعد أن يحرره، أى بعد ترك المستخدم للمفتاح أو بعد عملية الضغط.

رابعاً : أحداث التعامل مع النافذة :

١٢ - حدث تغيير الحجم Resize

يقع هذا الحدث كلما تغير حجم النافذة، وذلك سواء من المستخدم أو من خلال البرنامج نفسه، ويفيد هذا الحدث عندما تريد تغيير حجم الأدوات وذلك كلما تغير حجم النافذة.

```
Private Sub form1_resize()
        Text1.Move 0, 0, ScaleWidth, ScaleHeight
End Sub
```

حيث يقوم هذا الأمر بتغير حجم خانة النص Text1 والخاصة بأداة مربع النص على نافذة البرنامج ليجعلها بنفس حجم النافذة.

١٣ - حدث التحميل Load

هذا الحدث خاص بتحميل النافذة، أى يقع هذا الحدث عند تحميل النافــذة فــى الذاكرة وقبل إظهارها للمستخدم، وأيضاً بالنسبة للنافذة الرئيسية للبرنامج يقع عندما تبدأ فى تشغيل البرنامج.

١٤ - حدث إظهار النافذة Show

يقع هذا الحدث عندما يريد المستخدم أن تظهر نافذة ما علــى الشاشــة وذلــك فى حالة عدم وجودها من قبل فى الذاكرة، وبالتالى سيقع الحدث عنــدما ســتظهر النافذة على الشاشة.

مثال :

كتابة الأمر : Form1.Show

لإظهار نافذة البرنامج المسماة Form1 على الشاشة.

أما فى حالة التحميـــل للـــذاكرة للنافـــذة، فـــيمكن أن يـــتم تنفيـــذ الأمر التالى : Form1.Load

١٥ - حدث إخفاء النافذة Hide

هذا الحدث عكس الحدث السابق ويختلف هذا الحدث حسب حالة المستخدم.

على سبيل المثال : يريد المستخدم إخفاء النافذة من على الشاشة فقط مع بقاؤها فى الذاكرة، لذا يمكن استخدام الأمر :

```
Form1.Hide
```

مع العلم بأنه يمكن إظهارها مرة أخرى باستخدام الأمر Show أى :

```
Form1.show
```

هذا بالإضافة إلى أنه يمكن أن تستخدم خاصية الإخفاء الموجودة في نافذة الخصائص، بشرط أن تغيرها إلى حالة الحالة False، وذلك كما يلي :

```
Form1.Visible = False
```

وبالطبع من خلال نفس الخاصية يمكن التعديل لها بالإظهار كما يلي :

```
Form1.Visible = True
```

✍ **ملاحظة : (لا يتم تنفيذه)**

بالنسبة لإخفاء وإزالة النافذة من الذاكرة فيمكن استخدام الأمر :

```
Form1.Unload
```

١٦ - حدث الرسم **Paint**

يقع هذا الحدث عندما يحتاج المستخدم إلى إعادة رسم النافذة من جديد، على سبيل المثال حاجة المستخدم لتكبير النافذة، أو أن يغطى نافذة البرنامج بنافذة أخرى وبعد ذلك يعيد إظهارها مرة أخرى.

✍ **ملاحظة :**

يجب أن تكون خاصية إعادة الرسم تلقائيا AutoRedraw في الوضع False.

١٧ - حدث غلق النافذة **QueryUnload**

هذا الحدث خاص بالاستفسار عن غلق النافذة، حيث يقع عندما يريد المستخدم غلق النافذة، وبالطبع أنت فى حاجة لذلك عندما تحتاج لعرض أى رسائل أو نتائج قبل إنهاء النافذة.

٣ – الجزء الخاص بكتابة الأوامر وللتعليمات (البرامج الفرعية) :

حيث ترتبط بالحدث أو الأداة وستظهر شاشة الكتابة للتعليمات المختلفة الخاصة بالأداة المراد تنفيذها فيما بعد.

تبدأ هذه الخانة بكلمتى Private Sub وتنتهى بكلمة End Sub وفيما بينهما يمكن كتابة التعليمات أو الأوامر المراد تنفيذها، (وتسمى سطرى الإجراء) كما سيظهر من الشكل التالى :

```
Form                    Load
Private Sub Form_Load()

End Sub
```

أى أن هذه الشاشة مخصصة لكتابة التعليمات أو الأوامر خاصة بالبرنامج أو الحدث الذي سيتم تنفيذه.

حيث يتم كتابة الأوامر بطريقة متتالية بدون تحديد أرقام للأسطر مثل البيسيك القديم، مع الضغط على مفتاح التشغيل Enter للانتقال للسطر التالي أو لكتابة الأمر التالي.

ويلاحظ أن كل حدث أو برنامج فرعى يكتب مستقلاً عن الأخر بفاصل خط بين كل برنامج وأخر، ويلاحظ أيضا أن كتابة التعليمات تقع بين كلمة بداية البرنامج وتعريفه Sub و كلمة نهاية البرنامج End Sub.

فى هذا الجزء يمكن تنفيذ وظائف الإلغاء والإدراج والانتقال باستخدام مفاتيح لوحة الإدخال Key Board العادية Del و Ins و الأسهم ...الخ.

٤- أشرطة التمرير الرأسية والأفقية:

بالطبع الهدف منها التحكم فى عرض أو ظهور محتويات شاشة البرنـــامج (القسم الثالث السابق) من الأوامر والتعليمات، حيث يتم الـــتحكم فـــى عـــرض محتوياتها بطريقة رأسية أو أفقية، وذلك بالضغط على السهم حسب الأتجاه الـــذى تريده، حتى يمكنك القيام بأى عمليات تعديل أو كتابة جديدة للتعليمات أو الأوامر.

مثال علم للسابق :

```
Private Sub Label1_Click()
        Label1.Caption = CurDir$
End Sub
```

وهذا الأمر خاص بمعرفة اسم الدليل النشط الحالى، حيث تم اختيار أداة عنوان ووضعها على نافـــذة البرنـــامج، ثـــم تـــم فـــتح شاشـــة الكـــود الخاصـــة بهـــا وكتابة الأمر : Label1.Caption = CurDir$ كما سبق.

شرح المثال السابق :

- السطر الأول :

حيث تشير كلمتى **Private Sub** إلى برنامج فرعى خـــاص باســـم الأداة التالية له، ويتم كتابة هذا السطر تلقائياً فور الضغط المـــزدوج علـــى الأداة، حيث تعتبر كلمة Label1 هى الأداة المراد كتابة التعليمات لها.

وكلمة Click()_ تشير للحدث المطلوب وهو النقر عليها لتنفيذها.

- السطر الثانى :

يشير للأمر المراد تنفيذه عند تنفيـــذ الحـــدث أى عنـــد النقـــر علـــى الأداة المسماة Label1.

وتشير **Caption** إلى خاصية العنوان.

وتشير كلمة **CurDir$** إلى عرض اسم الدليل الحالى الجارى التعامل معه.

- السطر الثالث :

تشير كلمتى **End Sub** إلى نهاية البرنامج الفرعى المسئول عن تنفيذ هــذه الأداة، ويتم كتابة هذا السطر تلقائياً فور الضغط المزدوج على الأداة.

*** مثال تجميعى لجميع الإجراءات.**

بفرض أنك تريد تصميم شاشة تحتـوى علـى عـدة مفـاتيح منهـا مفتـاح أو زر خاص بطباعة Print رسالة معينة مثل (جامعة المنصورة - كلية التربيـة النوعيـة بـدمياط)، وأخـر خـاص بإعطـاء أو إصـدار صـوت Beep، وأخر خاص بإنهاء البرنامج أو الخروج منه.

لتنفيذ ذلك يمكن إتباع الآتي :

١- بالطبع تشغيل برنامج VB، وسيمنحك البرنامج نوافذ جديدة أو يمكنـك فـتح نافذة File، واختيار أمر مشروع جديد New Project.

٢- عندئذ ستظهر نافذة برنامج جديدة (كل النوافذ جديدة).

٣- البدء فى تصميم المفاتيح أو الأزرار لكتابة بياناتها:

Print Sound Exit

كل على حدة من خلال أداة زر الأمر CommandButtom وبالطبع الكتابة لكل كلمة فى خاصية Caption لكل زر.

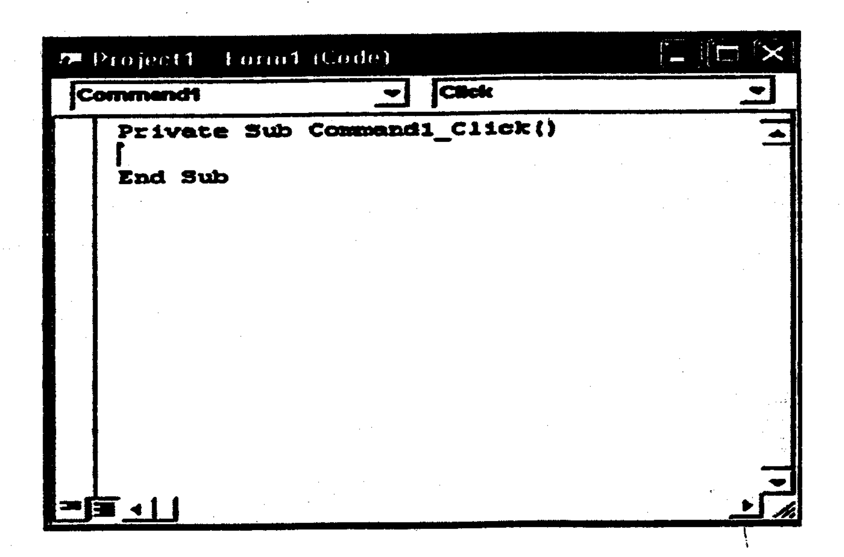
٤- يتم ذلك من خلال تحريك المؤشر إلى نافـذة صـندوق الأدوات ToolBox وتحريك المؤشر على رمز الزر الخاص بإضـافة CommandButtom، ثم النقر مرة واحدة على المفتاح الخاص بهـا، ثـم الانتقـال إلـى نافـذة البرنامج Form1، ثم رسم الأداة فى المكان الذى تريده، ثم الكتابة فى نافـذة الخصائص Properties فى الخاصية Caption كلمة Print.

٥- عندئذ يمكن كتابة التعليمات الخاصة بهذه الأداة (البرنامج الفرعى الخـاص بتشغيلها) من خلال :

أ - تحريك المؤشر على هذه الزر والنقر مزدوج عليها.

ب- تنشيط نافذة الكود Code من خلال النافذة الفرعية المسماة View.

٦ - عندئذ ستظهر نافذة الكود بالشكل التالى :



عندئذ يمكن كتابة الرسالة التى تريد طباعتهـا مـن خـلال الضـغط علـى مفتاح Print بالشكل التالى :

كتابة كلمة Print فى مود Mode اللغة الإنجليزية العادى، ثم الضغط علـى مفتاحي Alt + Right Shift، والكتابة للرسالة التـى تريـدها باللغـة العربيـة محصورة بين علامتي تنصيص " " وبالتحديد بين سطرى الأجراء وهى :

Print "جامعة المنصورة - كلية التربية النوعية بدمياط"

وال Project و Code و...و Color

٧ - سنلاحظ فى نافذة البرنامج وجود كلمة Click فى خانة الإجراء (الحدث) وكلمة Command1 فى خانة Object الخاصة باسم الأداة، وهذا وضع افتراضي منحه البرنامج نتيجة ترتيب التنفيذ كما سبق:

```
Private Sub Command1_Click()
    Print " جامعة المنصورة – كلية التربية النوعية بدمياط "
End Sub
```

٨ - وهكذا ستتغير محتويات خانة الأداة تبعاً لرقم الأداة، وهنا يتم استكمال إضافة باقى الأدوات للنموذج،

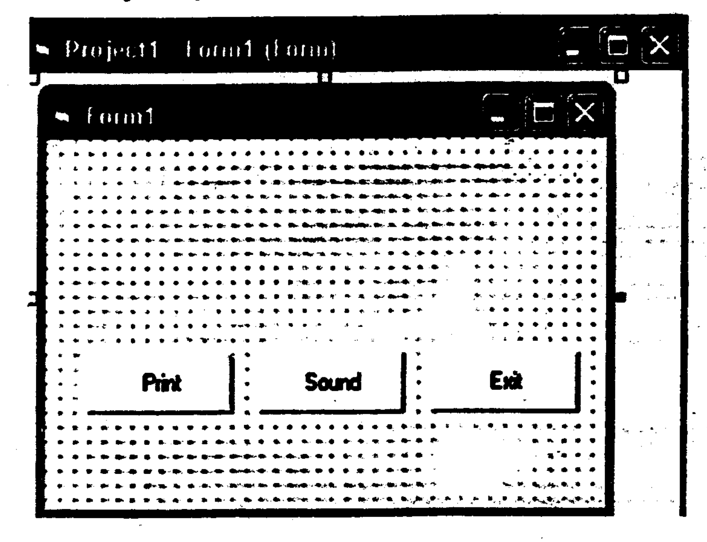
- فيتم وضع الكود التالى فى كود الزر Beep:

```
Private Sub Command2_Click()
    Beep
End Sub
```

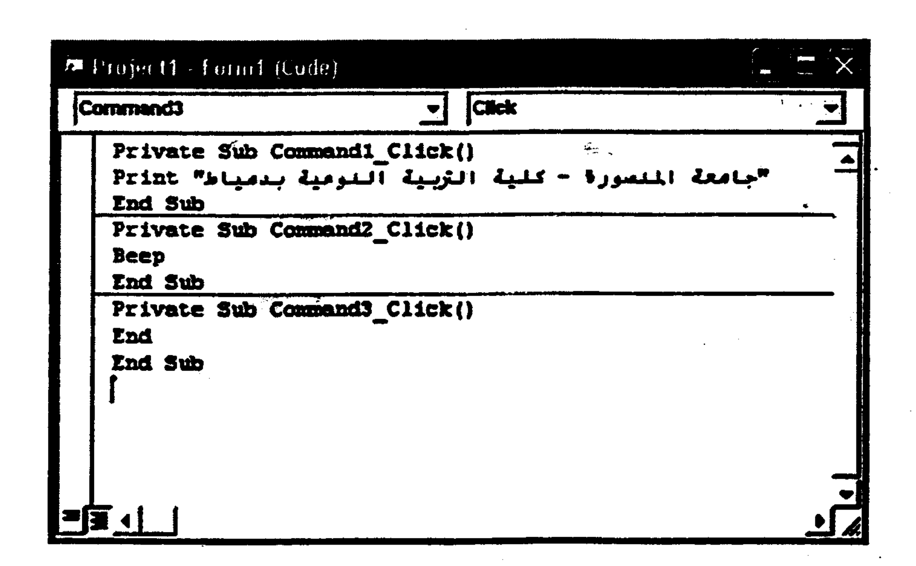
- والكود التالى فى كود الزر Exit:

```
Private Sub Command3_Click()
    End
End Sub
```

وفيما يلى شكل نافذة البرنامج بعد تصميمها (أثناء مرحلة التصميم) :

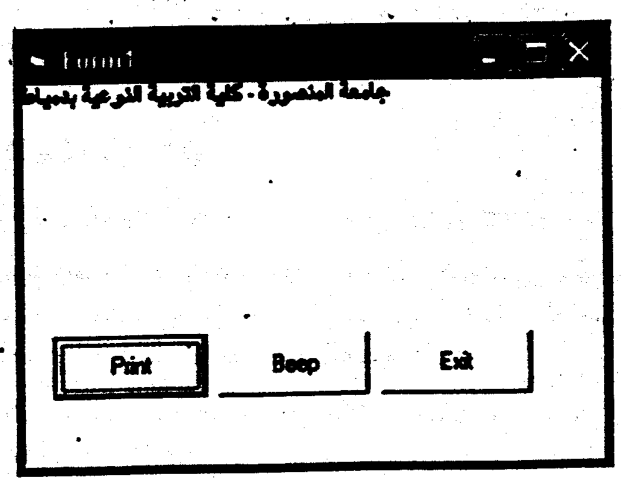


وفيما يلى شكل نافذة الكود الخاصة بالنافذة السابقة



نوافذ Project و Code وبعد Color

وفيما يلى شكل نافذة البرنامج بعد تنفيذه فى مرحلة التنفيـذ (التشـغيل) للبرنامج:



✍ ملاحظة :

يمكن تغيير أسماء الأدوات السابقة:
(Command1 , Command2 , Command3) والموضوعة على نافذة البرنامج المسماة Form1 الاسم الذى نوعه من خلال نافذة الخصائص وبالتحديد مـن خـلال خاصـية Name عـند توصـيف كل أداة لكتابة البرنامج الفرعـى الخـاص بهـا فـى نافـذة الكـود، وكذا بالنسبة لنافذة البرنامج نفسها.

ثم يجب الالتزام بالاسم الجديد فى نافذة الكود بناءاً علـى القاعـدة السابق تنفيذها.

* تنفيذ (تشغيل Run) البرنامج :

وللتنفيذ هذا البرنامج لمشاهدة نتيجة التنفيذ يمكن التنفيذ من خلال الآتي :

الضغط على F5 من لوحة المفاتيح أو الضغط على زر السهم المتجه لليمين من شريط الأدوات أو فتح نافذة Run واختيار أمـــر Start، وعندئـــذ ســـيتم التنفيـــذ للمشروع بالكامل، وهنا ستظهر نافذة النموذج بدون وجود النقاط التى تظهر بهـــا أثناء مرحلة تصميم البرنامج.

بالنسبة لناتج التنفيذ للمثال السابق :

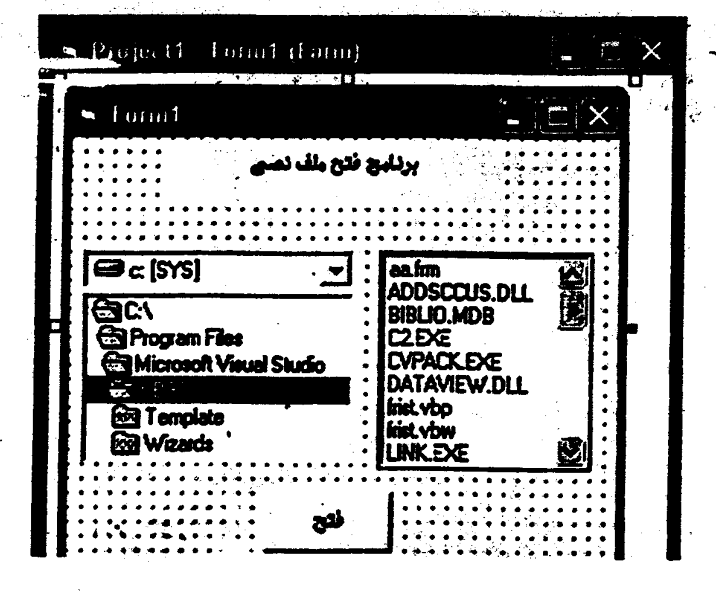
- عند التنفيذ وظهور شاشة التنفيذ (التى لا تحتوى على نقاط).
- يتم مثلا تحريك المؤشر على كلمة Print و النقر عليها من خلال الزر الأيسر للفأرة نقرة واحدة ستظهر على أعلى الشاشـــة مـــن ناحيـــة اليســـار رســـالة " جامعة المنصورة – كلية التربية النوعية بدمياط ".
- وفى حالة الضغط على كلمة End سيتم إنهاء التعامل مع البرنامج فى مرحلـــة التنفيذ والعودة إلى مرحلة التصميم للبرنامج.
- وبالنسبة لكلمة Beep ستسمع صوت عند الضغط عليها.

مثال آخر :

فى حالة الحاجة للتعامل مع وحدة معينة أو دليل أو ملفات معينة وذلـــك أثنـــاء تنفيذ البرنامج الذى تعمل معه .

بالتالي يجب أن تحدد أدوات خاصة بذلك على نافذة البرنـــامج، ثـــم تحديـــد الأوامر اللازمة لتشغيلها من خلال نافذة الكود المرتبطة بهذه النافذة (Form).

والتنفيذ العملى لذلك يتضح نافذة التصميم التالى :



من خلال النافذة السابقة يتضح وجوب توفر الآتى :

١ – البرنامج الفرعى الأول لأداة Drive:

خاص بحدث التغيير والخاصة بعرض الأدلة.

٢ – البرنامج الفرعى الثانى لأداة Dir:

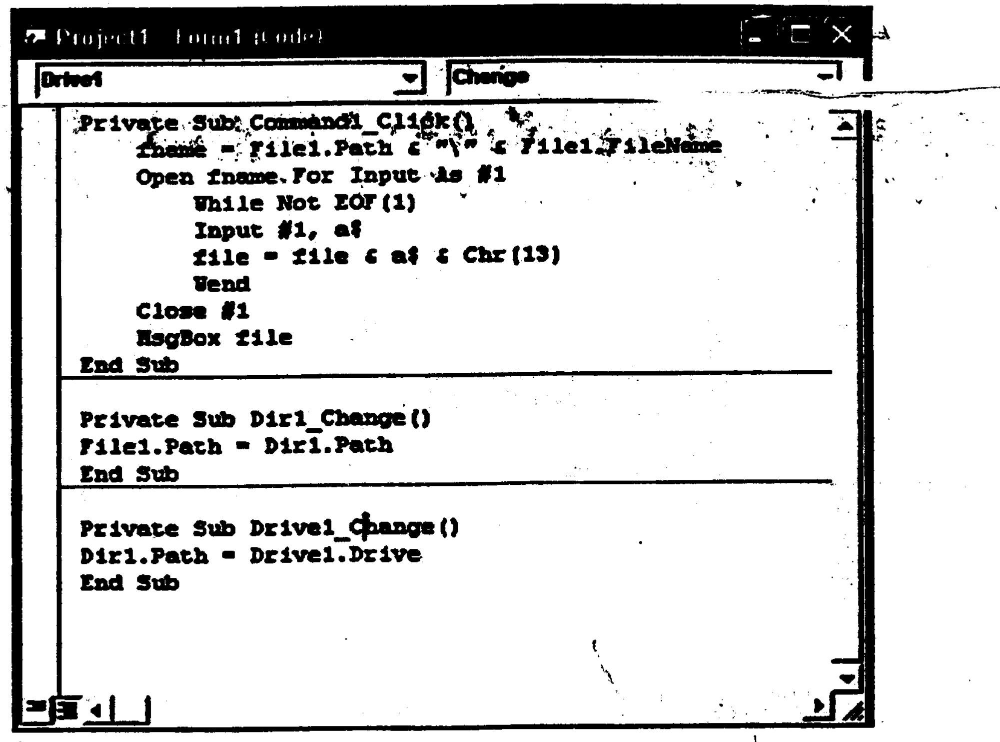
خاص بحدث التغيير والخاصة بعرض ملفات الدليل الحالى.

٣ – البرنامج الفرعى الثالث :

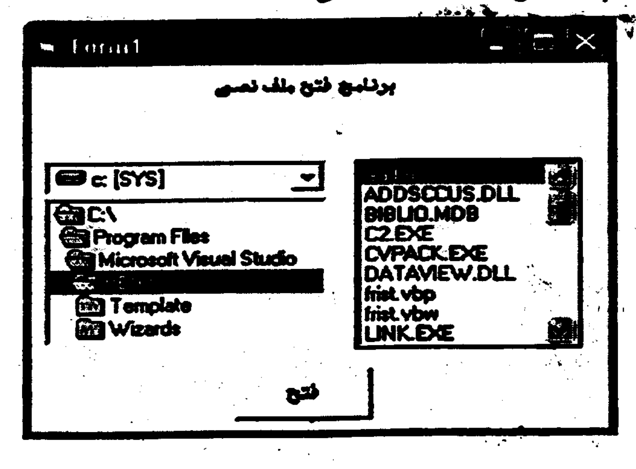
خاص بحدث النقر والذى يقوم بعرض محتويات الملف الذى تـم تحديـده بالأداة File.

والشكل التالى يوضح شكل الكود المطلوب :

وأخذ Project و Code و... Color



ناتج التنفيذ يتضح من الشكل التالى :



• الأدوات الغير مرتبطة بالحدث

من خلال هذا البرنامج VB تعاملنا من قبل مع بعض الأدوات التى ترتبط بأداة معينة ويمكن أن ترتبط بحدث معين مثل (النقر أو النقر المزدوج) لتنفيذه، ولكن هناك بعض الأدوات التى لا ترتبط بحدث معين وبالتالى لا يكتب لها تعليمات، مثل أداة الخط Line أو أداة الرسم الهندسى (الأشكال) Shape.

لأن هذه الأدوات تخصص للرسم على الشاشة، وبالتالى لا تستجيب هذه الأدوات لأى حدث يتم تنفيذه من قبل المستخدم على الشاشة.

وللتأكد من ذلك :

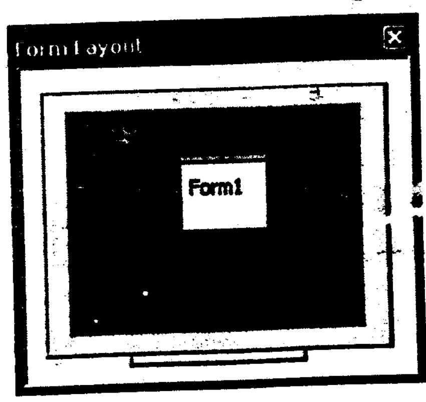
قم برسم أو وضع خط من خلال أداة الخط مثلاً على نافذة البرنامج Form، ثم قم بتنفيذ البرنامج، ستجد أن الخط موجود على شاشة البرنامج، وقم بتحريك مؤشر الفأرة عليه والنقر عليه، بالطبع لا يستجيب للحدث أو أى عمليات أخرى، معنى ذلك أن هناك بعض الأدوات لا تستجيب أو لا تشعر بالحدث.

٥ - ٣ نافذة التحكم فى موضع النموذج Form Layout Window

الغرض منها :

ضبط نافذة النموذج النشطة فى الموضع الذى يريد المصمم ظهورها به أثناء التنفيذ الفعلى للبرنامج، وذلك يساعد المصمم على التحكم فى المكان الذى سيظهر به برنامجه على الشاشة.

والشكل التالى يوضح شكل النافذة:



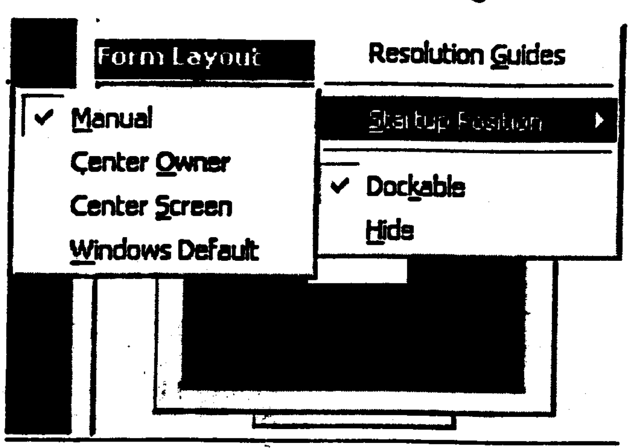
كيفية إظهارها :

تظهر هذه النافذة بشكل افتراضى مع البرنامج ولكن فى حالة عدم ظهورها أو بعد إخفائها يمكنك إعادة إظهارها مرة أخرى من خلال:

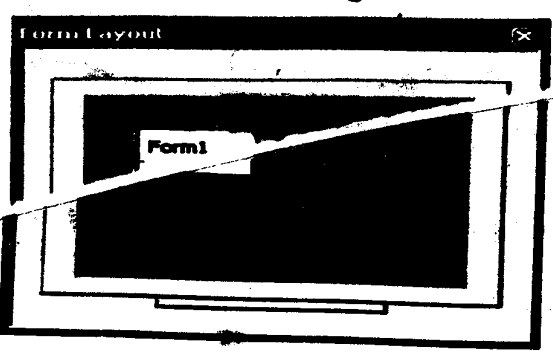
- فتح قائمة المشاهدة View.
- التحرك بالمؤشر لأمر Form Layout Window والنقر عليه.
- عندئذ ستظهر النافذة ويمكنك التحكم فى موضعها، مع العلم أن الموضع الإفتراضى لها هو أسفل يمين الشاشة حيث تتجمع ثلاث نوافذ افتراضية فى عمود واحد من خلال الخاصية الافتراضية Dockable.

كيفية استخدام نافذة التحكم بموضع النموذج :

- من خلال الضغط بزر الفأرة الأيسر على رمز النموذج.
- تحريك رمز النموذج للموضع المطلوب ظهوره فيه أثناء التنفيذ. أو :
- النقر بالزر الأيمن على رمز النموذج لتظهر قائمة مختصرة يمكن من خلالها اختيار الأمر Startup Position لتظهر قائمة فرعية يمكن من خلالها اختيار مواضع سابقة التعريف من خلال البرنامج، والشكل التالى يوضح شكل القائمة المختصرة :



- أما عند اختيار الأمر الأول وهو Resolution : يتم عرض بيان لحدود النافذة فى وضع 480*640 ووضع 600*800 أنظر الشكل التالى :



٥ - ٤ نافذة تحكم الألوان Color Palette :

هذه النافذة لا تعطى افتراضيا من برنامج اللغة.
ولكن يمكنك التحكم فى إظهارها عند تشغيل برنامج VB.

الغرض منها :

استخدام الألوان المختلفة مع البرنامج الذى يعده المستخدم من خلال VB لإعطاء شكل جمالى واستخدام أحسن ومريح لمستخدم البرنامج.

كيفية إظهارها :

- فتح قائمة المشاهدة View
- التحرك بالمؤشر لأمر Color Palette والنقر عليه لتنشيطه.
- عندئذ ستظهر قائمة رسم الألوان أسفل الشاشة، وبها مربعات مختلفة تحتوى على الألوان المدرجة فى برنامج النوافذ.

كيفية استخدام نافذة الألوان :

- من خلال نافذة البرنامج Form نقوم بوضع الأداة أو رسم الشكل عليها.
- تنشيط الأداة المراد تحديد اللون لها بالنقر عليها وظهور الصندوق ذو الإطار المحدد بالنقاط.
- الذهاب لنافذة الخصائص الخاصة بالأداة النشيطة وتعديل الخاصية المسماة Style إلى الاختيار **1- Graphical**

Color و ... و Code و Project تواقد

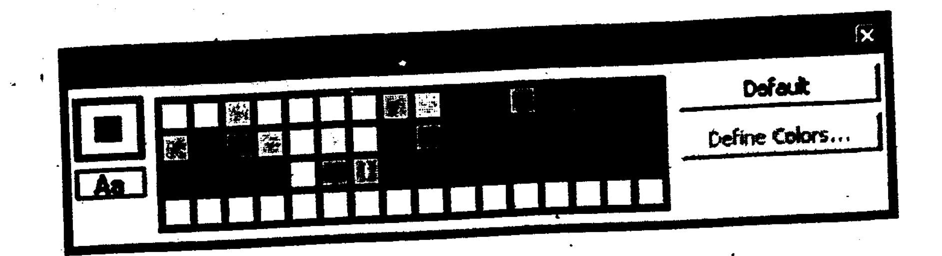
- ثم التحرك بالمؤشر علـــى نافــذة الألـــوان واختيـــار اللـــون المناســـب لـــك وذلك بالنقر عليه مره واحدة.
- بالتالى سنجد اللون الذى اخترته موجود على الأداة النشطة.

مثال :

إعطاء لون لأرضية نافذة البرنامج.

- فقط تنشيط النافذة بدون تنشيط أى أداة من الأدوات الموجودة عليها.
- ثم التحرك بالمؤشر على نافذة الألوان واختيار اللون المناسب لك وذلـــك بـــالنقر عليه مره واحدة.
- بالتالى سنجد اللون الذى اخترته موجود على أرضية نافذة البرنامج بالكامل بدون الأدوات التى بها.

وفيما يلى شكل هذه النافذة :



MR

الفصل السادس

الإجراءات والتفرع والتكرار

يتناول هذا الفصل المحتويات التالية :

- **الإجراءات**
- **التفرع والتكرار**

٦ - ١ - تعريف الإجراءات

هي برنامج مصغر أو هو جزء مستقل من البرنامج الرئيسي للمشكلة محل الدراسة أو برنامج فرعي، حيث يحتوى أيضا على مجموعة من الأوامر والمتغيرات والثوابت، ويشترك مع البرنامج فى أنه يمكن استدعاؤه باسمه (أى له اسم معين يمكن تشغيله به)، وعندما يتم تنفيذه من البرنامج العام يتم تنفيذ الأوامر الخاصة به، ثم بعد الانتهاء يعود إلى الأمر التالى له فى البرنامج العام لتنفيذه.

والسؤال لماذا تعد الإجراء ؟

والإجابة تتمثل فى حاجة البرنامج أو مخطط البرامج إلى تنفيذ أوامر معينة (مجموعة مترابطة من الأوامر تقوم بأداء وظيفة معينة لتحقيق غرض معين) أثناء تشغيل البرنامج العام (الرئيسى) فى مواضع معينة من البرنامج وبالتالى بصفة متكررة.

وبالتالى يمكن كتابة هذه الأوامر فى برنامج فرعى يسمى الإجراء مرة واحدة على أن يستدعى لتحقيق الغرض منه ثم يتم إنهاؤه. وهنا نستفيد بعدم تكرار أوامر متشابه فى البرنامج العام وأيضا حجم البرنامج يكون أصغر، وأيضاً لا يحتاج إلى ذاكرة كبيرة لتشغيل البرنامج الرئيسى.

بالإضافة إلى أنك فى حالة الحاجة لتعديل هذا البرنامج فأنك تقوم بالتعديل فيه بدلا من التعديل فى كل المواقع الخاصة بوجود هذه الوظيفة فى البرنامج العام.

٦ - ١ - ١ مكونات الإجراء :

تتمثل فى القاعدة التالية :

```
Sub Procedure name ( )
    Commands
    ------------
End sub
```


MR

الشرح لها :
السطر الأول : يشير إلى بداية الإجراء بـ Sub، ثم يأتى اسم الإجراء.
السطر الثانى : Commands تشير إلى الأوامر التى سيتم تنفيذها من خلال الإجراء (أمر يلى أمر).
السطر الثالث : يشير إلى نهاية الإجراء.

٦ - ١ - ٢ أنواع الإجراءات

١- الإجراء الخاص Private Procedure
٢ - الإجراء العام General Procedure

أولاً : بالنسبة للإجراء الخاص :
هو الاجراء الذى يرتبط بأداء وظيفة محددة يتم تنفيذها من خلال نافذة البرنامج Form أو من خلال أداة معينة، وبالتالى لا يمكن أن تستدعيه فى مكان أخر بالبرنامج العام.

على سبيل المثال :
الإجراء الخاص بأداة النص text التالية :

```
Private Sub Drive2_Change()
    Drive2.Height = 100
End Sub
```

مثال أخر :
لنسخ ملف معين من موقع إلى أخر

```
Private Sub Command1_Click()
    FileCopy "c:\demo\test1.frm ", "c:\dos\test1.exe'
End Sub
```

من خلال هذا المثال سوف يتم تخصيص زر باسم Command1 لتنفيذ أمر نسخ ملف والأمر هو Filecopy من مكان إلى أخر كما يظهر فى الأمر بالسطر الثانى.

ثانيا : الإجراء العام :

وهو أيضا يحتوى على مجموعة من الأوامر التى يمكن تنفيذها بعد استدعاؤها من خلال اسم الإجراء، مع العلم بأنه يمكن استدعاء هذا الإجراء من خلال أى موقع فى البرنامج لتنفيذه بالطبع لأنه إجراء عام لا يقتصر تنفيذه على موقع معين فى البرنامج.

يتبع نفس شروط الكتابة للإجراء الخاص ماعدا كتابة كلمة Private

على سبيل المثال :

```
Sub Getfilename ( )
        Commands
        -------------
End sub
```

من خلال هذا المثال :

يمكن تخصيص إجراء معين لفتح ملف معين أثناء تنفيذ البرنامج.
ثم تنفيذ الأوامر التى تريد التعامل معها.

مثال:

- فى حالة إعداد برنامج لحساب المرتبات للموظفين فى شركة ما، فيلزم قيام مخطط البرامج بإعداد برنامج فرعى لحساب التفقيط للصافى المستحق (أى كتابة قيمة المرتب بالأحرف) والتى تتغير هذه القيمة لكل موظف.

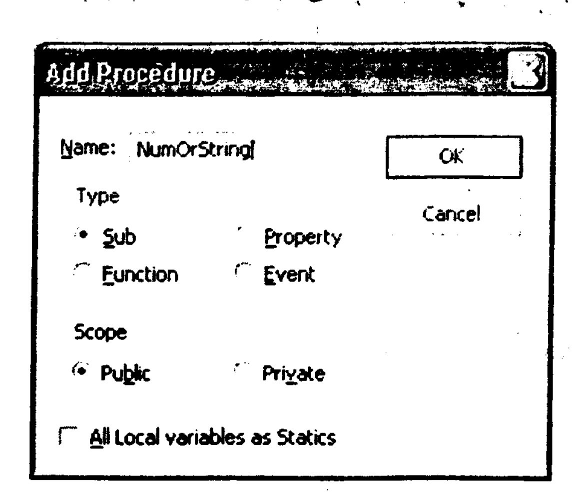
مثال آخر :

أثناء تشغيلك للبرنامج قد تحتاج فتح ملف بيانات معين وذلك لإضافة بيانات فيه أو عرض بيانات أو معلومات منه، وبالتالى يمكنك إعداد إجراء لاستدعاء ملف معين بكتابة اسمه من وبالتالى سيقوم الإجراء بفتحة فى مواقع معينة من البرنامج الخاص بك.

ويمكن تنفيذ ذلك عملياً من خلال الآتى :

١ - تنشيط نافذة البرنامج Form
٢ - وضع أداة CommandButton على النافذة السابقة.

٣- النقر المزدوج على الأداة لفتح نافذة الكود.
٤- فتح قائمة أدوات Tools. وتنشيط أمر الإجراء Add Procedure لتنفيذ إجراء عام والضغط عليه، وبالتالى ستظهر النافذة التالية :



٥- وبتحديد اسم الإجراء ونوع Sub، والنطاق Scope الخاص بهذا الاجراء Public عام، واختيار أمر موافق Ok.
٦- عندئذ ستظهر نافذة الكود الخاصة بهذا الإجراء خاليه، ويجب عليك كتابة الأوامر الخاصة فيها وذلك كما يلى:

```
Project1 - Form1 (Code)
(General)                    NumOrString

Public Sub NumOrString()
a = InputBox("[illegible]")

If IsNumeric(a) Then
        MsgBox ("لقد قمت بإدخال رقم")
Else:   MsgBox ("لم تقم بإدخال رقم")

End If
End Sub

Private Sub Form_Load()

End Sub
```

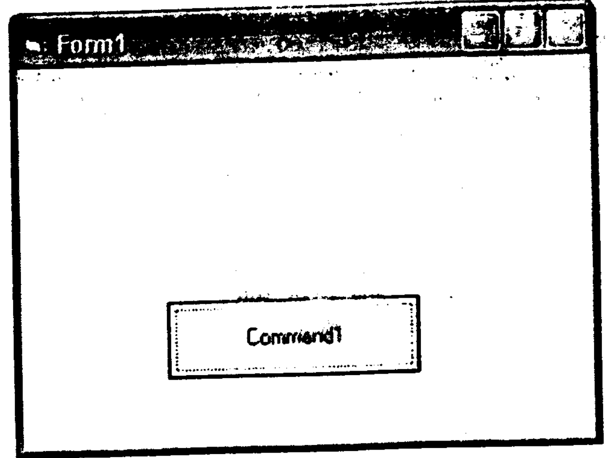
✍ ملاحظة :

من خلال تنفيذ أمر إدخال رسالة InputBox سيظهر صندوق يطلب منك إدخال رسالة أو قيمة، ومن خلال تنفيذ أمر إظهار رسالة MsgBox أيضاً سيظهر صندوق يقوم بعرض رسالة للمستخدم (عرض بيان معين).

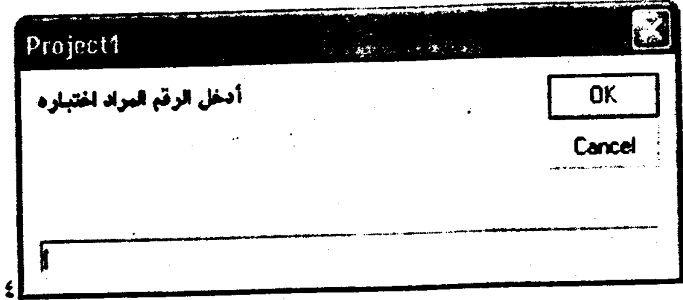
٧- ويمكن استدعاء الإجراء بكتابة اسمه فقط مع أمر Call أو في حالة توظيف أداة معينة مثل المثال الحالي يمكن الضغط على الشكل الخاص بها.

```
Private Sub Command1_Click()
    NumOrString
End Sub
```

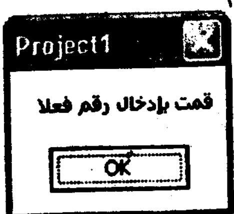
٨- وبالتالي يمكنك تنفيذ هذا الإجراء من خلال الترجمة التنفيذية له وذلك بالضغط على مفتاح F5، ليظهر الشكل التالي :



٩- وبالنقر علي زر الأوامر الخاص بهذه الأداة لتشغيله عندئذ ستظهر النافذة بالشكل التالي :



١٠- وبالطبع يمكنك الكتابة مباشرة من لوحة المفاتيح فى المربع السابق واختيار أمر موافق Ok لتنفيذ المطلوب، وفى حالة أن البيان الذى تم إدخاله هو رقم تظهر الرسالة التالية للمستخدم:



٦ - ١ - ٣ الإجراءات التى تعود بقيمة Function Procedure

الغرض منه :

هذا نوع أخر من الإجراءات يستخدم فى برامج البيسيك المرئى، حيث يقوم بحساب قيم معينة (حسابية) والعودة بهذه القيمة للمكان الذى تحدده فى البرنامج العام لإدخال هذه القيمة فى عمليات أو أوامر معينة.

شكل (مكونات) الإجراء

```
Function Procedure name ( )
        Commands
        --------------
End Function
```

شرح المكونات :

السطر الأول : تعريف الإجراء على أنه من نوع الـــFunction، مع تحديد اسمه.

السطر الثانى : يبدأ بالأوامر المراد تنفيذها فى هذا الإجراء.

السطر الثالث : يشير إلى نهاية الإجراء.

✍ ملاحظة :

فى حالة تمرير أو وضع متغيرات داخل هذا الإجراء يمكـــن أن تكتـــب الشكل التالى :

Function SumNum (A As Integer , B As Integer)

معنى ذلك أن هـــذا الإجـــراء ســـيحتوى علـــى متغيـــرات اســـمها A,B من النوع الصحيح Integer.

على سبيل المثال :

```
Private Function sumnum (a As Integer, b As Integer)
        sumnum = a + b ^2
End Function
```

٦ - ٤ مدى الإجراء :

يختلف هذا المدى طبقاً للمكان الذى يوجد به الإجراء، هـــل هـــو فـــى نافـــذة برنـــامج Form معينة أو هل يوجد فى برنامج عام Module (ذات النوع Bas).

ففى حالة تصميم الإجراء فى موقع ما على نافذة برنامج، فالطبع سيكون المدى لهذا البرنامج هو نطاق النافذة فقط.

وأما فى حالة وضعك لهذا الإجراء فى برنامج كود Module مثلاً سيكون مدى هذا الإجراء عام الاستخدام (أى سيكون متاح لك فى كل مواقع البرنامج) ويمكنك استدعاؤه فى أى مكان.

٦ - ١ - ٥ مكان الإجراءات :

يمكن وضع الإجراء فى الأماكن التالية :

١ - على نافذة البرنامج Form

التنفيذ بالطبع من خلال الآتى:

- تنشيط النافذة أو الأداة.
- النقر المزدوج عليها.
- سيظهر نافذة الكود الخاصة بها.
- كتابة التعليمات الخاصة به فى هذه النافذة.
- حفظ الإجراء.

٢ - فى برنامج Module

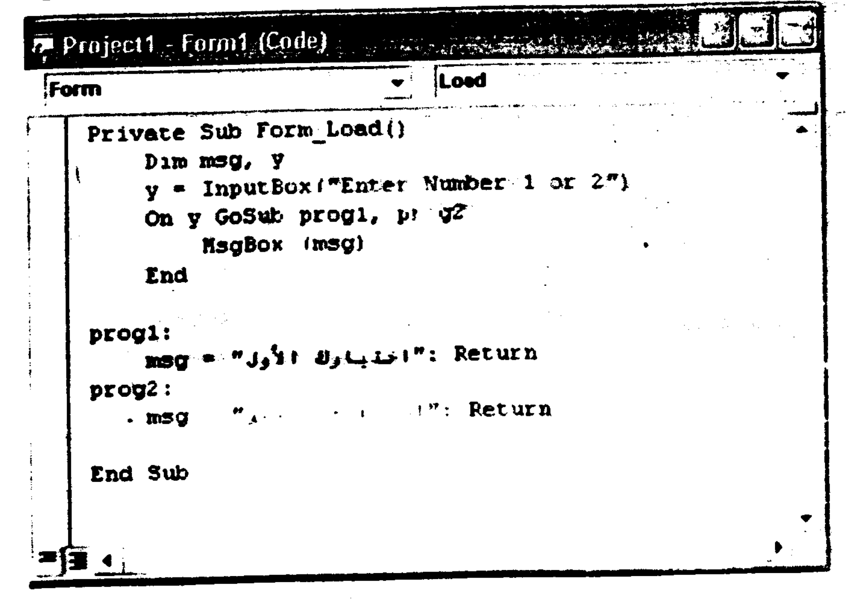
التنفيذ بالطبع من خلال الآتى :

- فتح نافذة Project، اختيار أمر Add Module.
- كتابة التعليمات الخاصة به فى هذه النافذة.
- حفظ الإجراء.

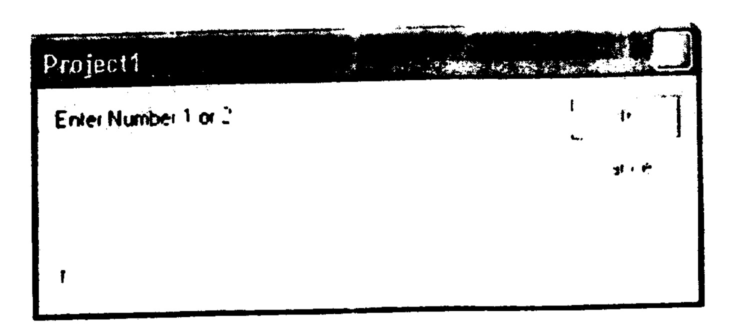
٦ - ١ - ٦ كيفية التعديل فى برنامج الإجراء:

لتنفيذ ذلك لابد من إتباع الآتي :

- تنشيط نافذة البرنامج Form الذى تم حفظ الإجراء فيها.
- البحث فى نافذة الكود الخاصة بهذه النافذة عن اسم الإجراء وتنشيطه.



وبالطبع لتنفيذ البرنامج يمكن الضغط على مفتاح F5 وبالتـــالى ســتظهر النافــذة او الصندوق التالى



و هنا يمكنك ادخال قيمة ما ١ او ٢ ثم الضغط على مفتاح Enter أو زر OK.
وبالتالى سيظهر على الشاشة الاجابة التالية :



✍ ملاحظة :

يمكن تغيير اسم المشروع بدلا من Project1 الاسم الافتراضي إلى test1، وسيظهر ناتج التنفيذ كما يلي:



الفصل السابع

تدريبات عملية

المحتويات

التدريب الأول

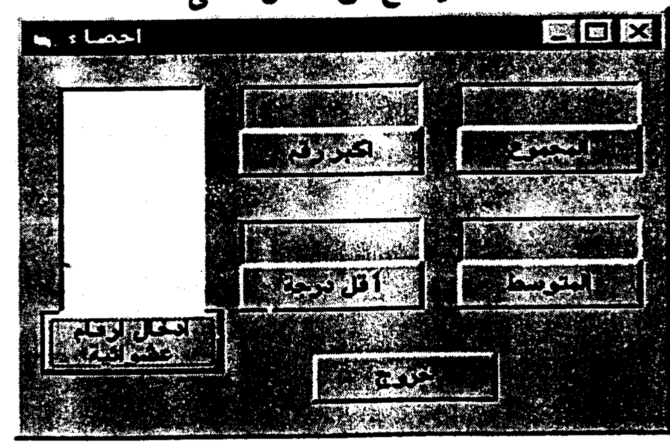
تعبئة قائمة من الأرقام العشوائية تتراوح بين ١٠٠ و ١٠٠٠ ثم إيجاد مجموعها ومتوسطها واكبر واقل رقم فيها

أولا : الأدوات المطلوبة

يتم إضافة الأدوات التالية على نموذج جديد :

- عدد ٦ زر أمر
- عدد ٤ مربع عنوان
- عدد ١ قائمة عرض

كما يتضح من الشكل التالى



ثانياً : الأكواد

فى حالة تحميل النافذة

```
Private Sub Form_Load()
    Caption = "احصاء"
End Sub
```

عنوان النافذة

فى حالة النقر على زر إدخال أرقام عشوائية

```
Private Sub cmdrnd_Click()
```

تخزين عدد الأرقام العشوائية المطلوبة فى متغير

```
n = Val(InputBox("أدخل عدد الارقام العشوائية"))
```

تنظيف القائمة من أى أرقام سابقة ان وجدت

```
lstrnd.Clear
```

بداية حلقة تكرارية بعدد الأرقام العشوائية

```
For i = 1 To n
```

تخزين رقم عشوائى يتراوح بين ١٠٠ و ١٠٠٠ داخل متغير

```
x = Int(100 + (1000 - 100) * Rnd)
```

إضافة الرقم العشوائى فى القائمة

```
lstrnd.AddItem x
```

نهاية الحلقة التكرارية وتعنى اعادة التحكم لبداية الحلقة

```
Next
End Sub
```

فى حالة النقر على زر المجموع

```
Private Sub cmdsum_Click()
```

الخروج من الإجراء إذا كان عدد عناصر القائمة صفر

```
If lstrnd.ListCount = 0 Then Exit Sub
```

تحديد متغير رقمى كمخزن للمجموع

```
Sum = 0
```

بداية حلقة تكرارية بعدد عناصر القائمة

```
For i = 0 To lstrnd.ListCount - 1
```

زيادة متغير المجموع بقيمة العنصر التالى بالقائمة

```
Sum = Sum + lstrnd.List(i)
```

نهاية الحلقة التكرارية وتعنى إعادة التحكم لبداية الحلقة

```
Next
```

عرض المجموع داخل المربع المخصص له

```
lblsum.Caption = Sum
End Sub
```

فى حالة النقر على زر المتوسط

```
Private Sub cmdaverage_Click()
```

الخروج من الإجراء اذا كان عدد عناصر القائمة صفر

```
If lstrnd.ListCount = 0 Then Exit Sub
```

تحديد متغير رقمى كمخزن للمجموع

```
Sum = 0
```

بداية حلقة تكرارية بعدد عناصر القائمة

```
For i = 0 To lstrnd.ListCount - 1
```

زيادة متغير المجموع بقيمة العنصر التالى بالقائمة

```
Sum = Sum + lstrnd.List(i)
```

نهاية الحلقة التكرارية وتعنى إعادة التحكم لبداية الحلقة

```
Next
```

عرض المتوسط (مجموع القيم على عددها) داخل المربع المخصص له

```
lblaverage.Caption = Sum / lstrnd.ListCount
End Sub
```

فى حالة النقر على زر اكبر رقم

```
Private Sub cmdmax_Click()
```

الخروج من الإجراء إذا كان عدد عناصر القائمة صفر

```
If lstrnd.ListCount = 0 Then Exit Sub
```

تعبئة متغير اكبر درجة بقيمة أول عنصر فى القائمة

```
mmax = lstrnd.List(0)
```

بداية حلقة تكرارية بعدد عناصر القائمة بدأ بثانى عنصر فى القائمة

```
For i = 1 To lstrnd.ListCount - 1
```

تعبئة متغير أكبر درجة بقيمة العنصر التالى بالقائمة وذلك فى حالة اذا كان المتغير أقل من قيمة العنصر التالى بالقائمة

```
If mmax < lstrnd.List(i) Then mmax = lstrnd.List(i)
Next
```

عرض اكبر رقم فى المربع الخاص بذلك

```
lblmax.Caption = mmax
End Sub
```

فى حالة النقر على زر اقل رقم

```
Private Sub cmdmin_Click()
```

الخروج من الإجراء إذا كان عدد عناصر القائمة صفر

```
If lstrnd.ListCount = 0 Then Exit Sub
```

تعبئة متغير اقل درجة بقيمة أول عنصر فى القائمة

```
mmin = lstrnd.List(0)
```

بداية حلقة تكرارية بعدد عناصر القائمة بدأ بثانى عنصر فى القائمة

```
For i = 1 To lstrnd.ListCount - 1
```

تعبئة متغير أقل درجة بقيمة العنصر التالى القائمة وذلك فى حالة اذا كان المتغير أكبر من قيمة العنصر التالى بالقائمة

```
If mmin > lstrnd.List(i) Then mmin = lstrnd.List(i)
Next
```

عرض اكبر رقم فى المربع الخاص بذلك

```
lblmin.Caption = mmin
End Sub
Private Sub cmdend_Click()
```

الخروج من البرنامج

```
End
End Sub
```

التدريب الثانى

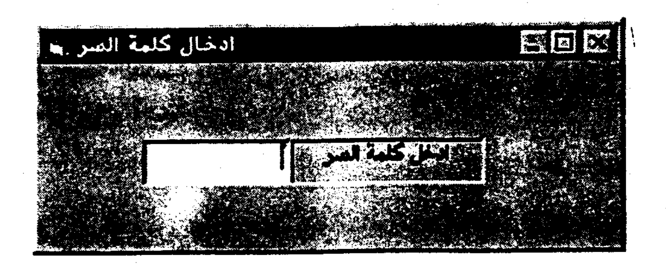
عمل نافذة للتأكد من صحة إدخال كلمة السر المدخلة عن طريـق المستخدم ومطابقتها لكلمة السر الموجودة بالملف الموجـود داخـل فهرس النظام لويندوز٩٨

أولا : الأدوات المطلوبة

يتم إضافة الأدوات التالية على نموذج جديد :

- عدد ١ مربع عنوان
- عدد ١ مربع نصى

كما يتضح من الشكل التالى



ثانياً : الأكواد

فى حالة تحميل النافذة

Private Sub Form_Load()

عنوان النافذة

Caption = "ادخال كلمة السر"

End Sub

عند الضغط على أى مفتاح داخل مربع إدخال كلمة السر

Private Sub txtpass_KeyPress(KeyAscii As Integer)

عند الضغط على مفتاح الإدخال (كوده ١٣)

If KeyAscii = 13 Then

الخروج من الإجراء فى حالة عدم وجود الملف الموجود به كلمة السر دخل فهرس النظام بالويندوز

If Dir$("c:\win98\system\vbwind.ocx") = "" Then Exit Sub

فتح الملف تتابعى من على فهرس النظام داخل الويندوز لقراءة كلمة السر

Open "c:\win98\system\vbwind.ocx" For Input As #1

تخزين كلمة السر الموجودة بالملف داخل متغير

Input #1, mpass

إغلاق الملف

Close #1

إذا كانت كلمة السر المدخلة عن طريق المستخدم تساوى كلمة السر الموجــودة بالمتغير

If mpass = txtpass.Text Then
MsgBox "كلمة السر صحيحة"

وإلا

Else
MsgBox "كلمة السر غير صحيحة"
End If

End If
End Sub

التدريب الثالث

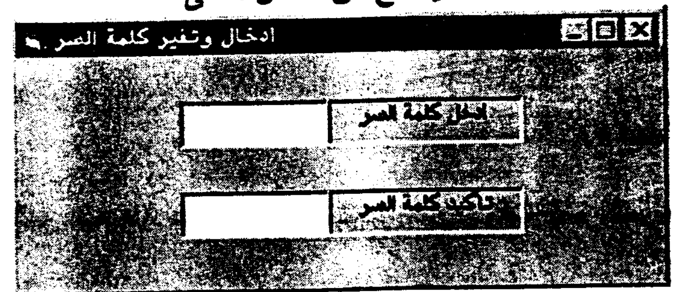
عمل نافذة إدخال وتغيير كلمة السر وتخزينها داخل ملف تتابعى داخل فهرس النظام لويندوز٩٨

أولا : الأدوات المطلوبة

يتم إضافة الأدوات التالية على نموذج جديد :

- عدد ٢ مربع عنوان
- عدد ٢ مربع نصى

كما يتضح من الشكل التالى



ثانياً : الأكواد

فى حالة تحميل النافذة

Private Sub Form_Load()

عنوان النافذة

Caption = "إدخال وتغير كلمة السر"

End Sub

عند الضغط على أى مفتاح داخل مربع إدخال كلمة السر

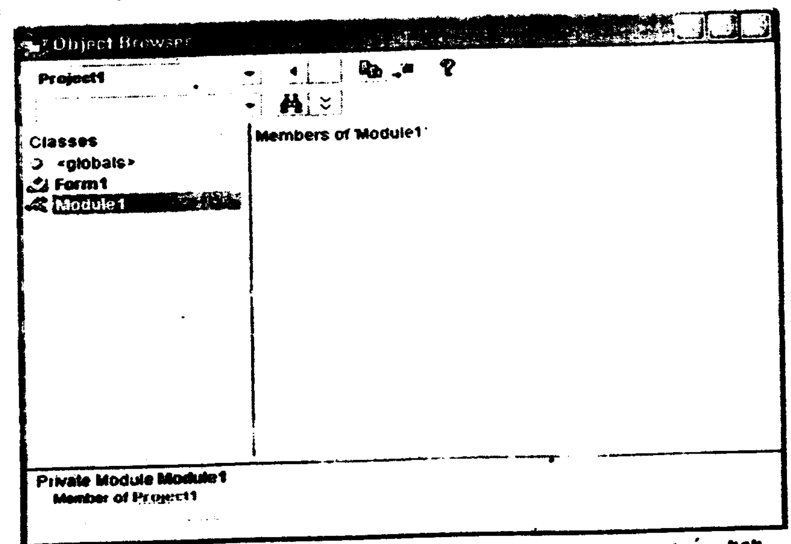
Private Sub txtpass_KeyPress(KeyAscii As Integer)

- عندئذ يمكن فتح خانة Object واختيار General منها ثم فتح القائمة الخاصة بالأحداث Events ومن خلال شريط التمرير يمكن البحث عن اسم الإجراء والنقر عليه لفتحة. وبالتالى سيظهر أوامر الإجراء أمامك، وبالتالى يمكن إجراء التعديل الذى تريده.

طريقة أخر :

من خلال الضغط على مفتاح F2 من لوحة المفاتيح :

- عندئذ سيظهر على الشاشة نافذة جديدة Object Browser.
- ومن خلالها يمكن اختيار اسم المشروع الحالى المراد التعامل معه فتظهر محتوياته من نماذج وModules والتى تتضح من الشكل التالى :



- وبالتالى يمكن اختيار اسم الإجراء بتحريك المؤشر عليه.
- ثم من خلال النقر المزدوج يمكن عرض محتوى هذا الإجراء من الأوامر والتعليمات.

٦ - ٢ عمليات التكرار واتخاذ القرارات

٦ - ٢ - ١ التكرار Looping :

الغرض منه :

تكرار تنفيذ مجموعة من التعليمات أو الأوامر عدد معين من المرات بدلاً من تكرار كتابة هذه الأوامر أكثر من مره فى البرنامج.

التكرار له قيمة بداية يبدأ بها، وأيضاً له قيم نهاية تنتهي عندها (بعدد مرات التنفيذ).

تستخدم هنا نفس أوامر التكرار المستخدمة فى لغة البيسيك العادية للتذكرة مثل :

١ - جملة التكرار For.......Next

على سبيل المثال :

```
For c = 1 To 10
    Print c
    c = c + 1
Next c
```

٢ - جملة التكرار Do while ...... Loop

التنفيذ نفس جملة التكرار السابقة.
والقاعدة كما يلى :

```
Do while <Condition >          <الشرط>
    Commands                    الأوامر
    -------------
Loop                 نهاية الدائرة التكرارية
```

✍ ملاحظة :

هنا يقوم برنامج VB بتقييم الشرط، فإذا كان الشرط خاطئاً فلا يقوم بتنفيذ التكرار مطلقاً.

وإذا كان الشرط صحيحاً فسوف يقوم بتنفيذ مجموعة الأوامر التي توجد في الدائرة التكرارية، وهكذا حتى يكون الشرط خاطئاً فسوف يخرج من الدائرة التكرارية.

٣ - جملة التكرار Do .....Loop while

التنفيذ نفس جملة التكرار السابقة.

والقاعدة كما يلي :

بداية الجملة التكرارية	**Do**
الأوامر	**Commands**

نهاية الجملة	**Loop while <Condition >**

✍ ملاحظة :

يتم تنفيذ التكرار مرة ثم يختبر الشرط فإن كان الشرط خاطئاً فيقوم VB بوقف التكرار، وإن كان صحيحا فيقوم بتكرار مجموعة الأوامر مرة أخرى ثم يكرر مرة أخرى اختبار الشرط.

٤ - جملة التكرار Do .... Until

التنفيذ نفس جملة التكرار السابقة.

والقاعدة كما يلي :

	Do Until < Condition>
الأوامر	**Commands**

نهاية الدائرة التكرارية	**Loop**

✍ ملاحظة :

التكرار هنا يستمر طالما الشرط خاطئا وينتهى متى اصبح الشرط صحيحا.

٥ - جملة التكرار Do ..... Loop Until

التنفيذ نفس جملة التكرار السابقة.
والقاعدة كما يلى :

```
Do                              بداية الجملة التكرارية
    Commands                    الاوامر
    --------------
Loop Until    <Condition >      نهاية الجملة
```

✍ ملاحظة :

التكرار هنا يستمر طالما الشرط خاطئا ويتوقف تنفيذ الشرط متى أصبح الشرط صحيحا

✍ ملاحظة عامة :

فى حالة الحاجة للخروج الاضطراري يمكن كتابة شرط معين من خلال الجملة التالية داخل الدائرة التكرارية :

```
For ........
    Commands
    IF <Variable > = "Stop" Then Exit For
Next ....
```

٦ - ٢ - ٢ الشروط (اتخاذ القرارات) :

ايضا راجع ذلك في لغة البيسيك العادية.
وفيما يلى أمثلة لذلك للتذكرة :

الشروط Conditions :

لا تخلو برامج الحاسبات من الشروط او اتخاذ القرارات داخل البرنامج، حيث يمكن من خلالها التحكم فى تنفيذ تعليمات معينة أو عدم تنفيذها.
ويتم ذلك من خلال تحقق شروط معينة أو عدم تحقق هذه الشروط (قد يكون شرط واحد).

على سبيل المثال قد يتعلق تنفيذ برنامج معين على تنفيذ شرط معين
والشرط هو تعبير منطقى إما أن يكون صحيح True أو خطا False.
وتستخدم فى اجراء تحقق الشروط عمليات المقارنة أو المعاملات المختلفة مثل المعاملات التالية :

- علامة أكبر من >
- علامة اصغر من <
- علامة أكبر من او يساوى >-
- علامة اصغر من أو يساوى <=
- علامة التساوى .
- علامة لا يساوى < >

والتنفيذ لذلك من خلال الجمل التالية .

١ جملة If........THEN

القاعدة العامة لها :

If < Condition > THEN Command

فى هذه الحاله فى حالة تحقق الشرط سيتم تنفيذ أمر واحد.
ملاحظه

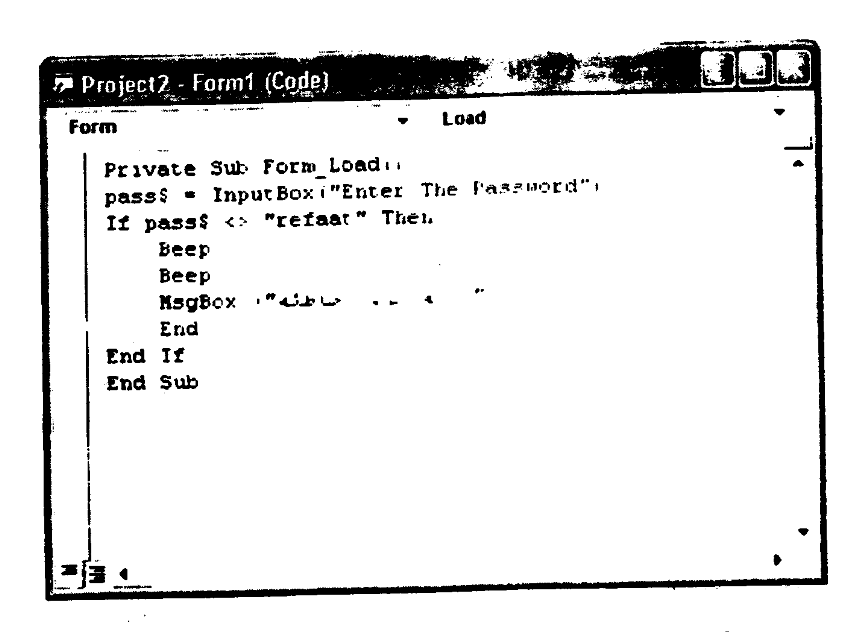
قد يكون هناك اكثر من امر(Commands) يتم تنفيذها فى حالـة تحقق الشرط، من خلال نفس القاعدة.

على سبيل المثال :

- فى حالة حساب قيم معينة داخل البرنامج مثل احسب أو ضع قيمة معينة لحقل معين فى سجل الملف الحالى (حساب الضرائب بنسبة ٣ فى المائة فى حالة أن المرتـب الأساسي أكبر من قيمة معينة).
- او فى حالة الحاجة لوضع كلمة سر لتنفيذ البرنامج الخاص بك.

هنا يجب تحقق شرط معين لدخول او فتح البرنامج، وهو معرفتك لهذه الكلمة لأنـه سيتم إجراء اختبار لها فى حالة تحققه يتم تشغيل البرنامج والعكس صحيح.

مثال :



من خلال هذا البرنامج (الاجراء الخاص) يحتوى على التعليمات الآتية بالترتيب

١ - بداية الإجراء الخاص بتحميل نافذة برنامج.
٢ - تم تحديد متغير حرفى باسم passS ليتم فيه إظهار صندوق إدخال لقيمة معينة للرسالة التالية **Enter The Password**.
٣ - لو أن القيمة المحجوزة فى هذا المتغير لا تساوى كلمة **refaat**) عندئذ :
٤ - نفذ أمر إخراج صوت **Beep** مرتين متتاليتين.
٥ - ثم اظهر صندوق لرسالة "كلمة مرور خاطئة" .
٦ - إنها التعامل **End**.
٧ - إغلاق دالة **IF**.
٨ - نهاية الإجراء الخاص.

ناتج التنفيذ :

ظهور النافذة التالية، وبالتالى يتم كتابة البيان المراد إدخالـــه وهـــو كلمـــة الســـر أو المرور:

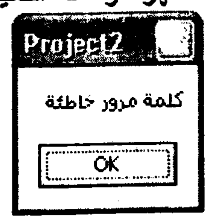
Project2

Enter The Password

OK

Cancel

وعند إدخال كلمة مرور خاطئة تظهر الرسالة التالية:



٢ - جملة Else If...THEN

القاعدة العامة لها :

```
If < Condition > THEN
    Commands
    ---------------
Else
    Commands
    --------------
End If
```

من خلال هذه الجملة الشرطية فان برنامج VB سيقوم بتقييم الشرط، فإذا كان الشرط صحيحاً فإنه ينفذ مجموعة من الاوامر التى تأتى بعد كلمة THEN حتى كلمة Else، وإذا لم يتحقق الشرط (أى خطأ) فانه سوف ينفذ مجموعة الأوامر التى تلى كلمة Else حتى نهاية الجملة الشرطية End If.

مثال لذلك :

```
Project2 - Form1 (Code)
Form                    Load

Private Sub Form_Load()
pass$ = InputBox("Enter The Password")
If pass$ <> "refaat" Then
    Beep
    Beep
    MsgBox ("كلمة مرور خاطئة")
    End
Else: MsgBox ("كلمة مرور صحيحة")
End If
End Sub
```

من خلال هذا البرنامج يتضح الآتى :

- فى حالة عدم تحقق الشرط (بخلاف ذلك Else)، ستظهر رسالة : "كلمة مرور خاطئة" .
- أما فى حالة تحقق الشرط سوف يظهر صندوق وبه رسالة : " كلمة مرور صحيحة ". ويمكن أن يتبع ذلك أوامر أخرى خاصة ببداية تشغيل البرنامج.
- ثم انهاء التعامل مع البرنامج.

✍ ملاحظة :

من خلال الجمل السابقة يمكن تنفيذها فى حالة وجود أو تحقق احتمال واحد أو احتمالين للشرط الواحد.

٣ - جملة Select Case

القاعدة العامة للأمر :

```
Select Case  < Condition >
    Case      < الاحتمال الأول >
        Commands
    Case      < الاحتمال الثانى >
        Commands
    Case      < الاحتمال الثالث >
        Commands
    Case Else
        Commands
End Select
```

تستخدم هذه الجملة فى حالة وجود أكثر من احتمال لتحقق الشرط، حيث يبدأ برنامج VB فى تقييم الشرط فى بداية الجملة، ثم مقارنته مع الاحتمالات

المختلفة التى تذكر مع حالة أو كلمة Case حيث يوجد لكل حالة مجموعة من الاوامر الخاصة بها تنفذ فى حالة تنفيذ الشرط عليها أو وقوع الاحتمال.

وإذا لم تتطابق الاحتمالات المختلفة مع الشرط المحدد فى بداية الجملة فإنه سوف ينفذ مجموعة الأوامر التى تلى كلمة Case Else (وهذه الجملة اختيارية أى فى حالة عدم وجودها لا يتم تنفيذ أوامر الاحتمالات السابقة).

٤ - جملة التفرع ON GOTO أو ON GOSUB :

الغرض منها :

فى حالة تحقق شرط ما يمكن أن تتفرع أو تذهب إلى موقع أو مكان معين فى البرنامج لتنفيذ أوامر معينة والعودة مره أخرى للأمر التالى للسابق التنفيذ منه.

القاعدة :

ON *Expression* Goto *Destination list*

شرح القاعدة:

ON : تشير الى بداية الجملة.

Expression : تشير إلى تعبير رقمى معين سابق قبوله أو إدخاله فى متغير مثلاً.

GOTO : تشير إلى الانتقال إلى رقم سطر معين فى البرنامج.

ONSUB : تشير إلى انتقال التنفيذ إلى مكان برنامج فرعى.

Destination list : تشير الى قائمة بالعناوين أو أرقام السطور المراد التفرع أو الانتقال لها فى البرنامج، حيث يتم الفصل فيما بينها بفاصلة (،).

مثال لذلك :

فيما يلى برنامج للتفرع بناء على قبول قيمة معينة رقم ١ أو رقم ٢، وفى حالة الاختيار يمكن التفرع لتنفيذ برنامج فرعى معين خاص بكل اختيار:

الانتقال الى مربع تأكيد كلمة السر عند الضغط على مفتاح الإدخال

```
If KeyAscii = 13 Then txtpass2.SetFocus
End Sub
```

عند الضغط على أى مفتاح مربع تأكيد كلمة السر

```
Private Sub txtpass2_KeyPress(KeyAscii As Integer)
```

عند الضغط على مفتاح الإدخال

```
If KeyAscii = 13 Then
```

توجيه رسالة للمستخدم فى حالة عدم مطابقة كلمة السر بالمربعين والخروج من هذا الإجراء لعدم التطابق

```
If txtpass.Text<>txtpass2.Text Then
  MsgBox "كلمةالسر غيرمتطابقة" :Exit Sub
End if
```

فتح ملف تتابعى على فهرس فرعى داخل الويندوز حتى لا يعرف مكانه أحد

```
Open "c:\win98\system\vbwind.ocx" For Output As #1
```

تسجيل كلمة السر فى الملف

```
Write #1, txtpass.Text
```

إغلاق الملف

```
Close #1
```

توجيه رسالة إتمام الحفظ

```
MsgBox "تم حفظ كلمة السر"
```

تنظيف مربعات النصية من بياناتها

```
txtpass.Text = ""
txtpass2.Text = ""
```

الانتقال داخل مربع إدخال كلمة السر

```
txtpass.SetFocus

End If
End Sub
```

التدريب الرابع

إدخال بيانات طلبة الكلية ودرجات ثلاث مواد مع عرض مجموع هذه المواد والتقدير العام لكل طالب وعرض الطالب الأول على كل فرقة.

أولاً : الأدوات المطلوبة

يتم إضافة الأدوات التالية على نموذج جديد :

- عدد ٤ زر أمر
- عدد ٧ مربع عنوان
- عدد ٥ مربعات نصية
- عدد ٤ زر اختيار
- عدد ١ صندوق اختيار
- عدد ١ جدول

كما يتضح من الشكل التالى

بيانات طلبة

الرقم	الاسم	الفرقة	النوع	فيجول بيسك	ويندوز	اكسل	المجموع	التقدير
1	منى	الأولى	ذكر	100	49	49	198	امتياز
2	أحمد	الأولى	ذكر	90	40	30	160	جيد جداً
3	سلمى	الأولى	ذكر	75	25	30	130	جيد
4	ماجد	الأولى	ذكر	25	10	40	75	ضعيف
5	عمر	الأولى	ذكر	100	40	30	170	جيد جداً

اعرض المعلومات — اعرض الطالب الأول — خروج

ثانياً : الأكواد

الشرح	الكود
فى حالة تحميل النافذة	*Private Sub Form_Load()*
عنوان النافذة	*Caption= "بيانات طلبة"*
اختيار الفرقة الأولى من زر الاختيار	*optyear(0).Value = True*
اختيار النوع ذكر من صندوق الاختيار	*chkkind.Value = 1*
التحكم فى خصائص جدول عرض البيانات	
إخفاء جدول عرض البيانات لأنه خالى	*grid.Visible = False*
تحديد حجم الخط بالجدول	*grid.Font.Size = 10*
تحديد نمط الخط بالجدول	*grid.Font.Bold = True*
تحديد عدد الأعمدة بالجدول	*grid.Cols = 9*
تحديد عدد الصفوف بالجدول	*grid.Rows = 1*
الانتقال إلى الصف الأول لكتابة عناوين الأعمدة كما يلى	*grid.Row = 0*
الانتقال إلى العمود التاسع	*grid.Col = 8*
كتابة عنوان هذا العمود	*grid.Text = "الرقم"*
تحديد عرض هذا العمود	*grid.ColWidth(8) = 600*

ضبط اتجاه الكتابة فى هذا العمود ناحية اليمين

```
grid.ColAlignment(8) = 6
```

وهكذا لباقى الأعمدة

```
grid.Col = 7
grid.Text = "الاسم"
grid.ColWidth(7) = 1500
grid.ColAlignment(7) = 6

grid.Col = 6
grid.Text = "الفرقة"
grid.ColWidth(6) = 800
grid.ColAlignment(6) = 6

grid.Col = 5
grid.Text = "النوع"
grid.ColWidth(5) = 800
grid.ColAlignment(5) = 6

grid.Col = 4
grid.Text = "فجول بيسك"
grid.ColWidth(4) = 1000
grid.ColAlignment(4) = 6

grid.Col = 3
grid.Text = "ويندوز"
grid.ColWidth(3) = 800
grid.ColAlignment(3) = 6
```

```
grid.Col = 2
grid.Text = "الكسل"
grid.ColWidth(2) = 800
grid.ColAlignment(2) = 6

grid.Col = 1
grid.ColWidth(1) = 800
grid.Text = "المجموع"
grid.ColAlignment(1) = 6

grid.Col = 0
grid.ColWidth(0) = 800
grid.Text = "التقدير"
grid.ColAlignment(0) = 6
End Sub
```

حفظ بيانات طالب داخل ملف البيانات عند النقر على زر الحفظ

```
Private Sub cmdsave_Click()
```

الخروج من إجراء الحفظ فى حالة نقص البيانات

```
If txtno.Text = "" Or txtname.Text = "" Or txtvisual.Text = ""
Or
txtwindows.Text = "" Or txtexcel.Text = "" Then
  MsgBox "استكمل بيانات الطالب" :Exit Sub
End if
```

إذا كان ملف البيانات غير موجود يتم إنشاءه والكتابة فيه

```
If Dir$("data") = "" Then
 Open "data" For Output As #1
Else
```

إذا كان ملف البيانات موجود يتم فتحه للإضافة فيه

```
 Open "data" For Append As #1
End If
```

تعبئة متغير الفرقة بفرقة الطالب التى تم اختيارها من زر الاختيار

```
If optyear(0).Value = True Then myear = "الأولى"
If optyear(1).Value = True Then myear = "الثانية"
If optyear(2).Value = True Then myear = "الثالثة"
If optyear(3).Value = True Then myear = "الرابعة"
```

تعبئة متغير الحالة بكلمة ذكر إذا تم اختيارها من صندوق الاختيار وإلا تعبئة المتغير بكلمة أنثى

```
If chkkind.Value = 1 Then mkind = "ذكر" Else mkind = "انثى"
```

حفظ البيانات داخل الملف

```
Write #1, txtno.Text, txtname.Text, myear, mkind, txtvisual.Text,
txtwindows.Text, txtexcel.Text
```

اغلاق الملف

```
Close #1
```

تنظيف المربعات النصية عقب الحفظ

```
txtno.Text =""
txtname.Text =""
txtvisual.Text =""
txtwindows.Text =""
txtexcel.Text =""
```

الانتقال داخل مربع الرقم لإدخال طالب جديد

```
txtno.SetFocus
End Sub
```

عرض معلومات طلبة فرقة معينة عند النقر على زر العرض

Private Sub cmdshow_Click()

الخروج من إجراء عرض المعلومات فى حالة عدم وجود الملف

If Dir$("data") = "" Then MsgBox "الملف غير موجود" :Exit Sub

فتح الملف التتابعى للعرض

Open "data" For Input As #1

تعبئة متغير الفرقة بالفرقة التى اختارها المستخدم لعرض طلابها

If optyear(0).Value = True Then my = "الأولى"
If optyear(1).Value = True Then my = "الثانية"
If optyear(2).Value = True Then my = "الثالثة"
If optyear(3).Value = True Then my = "الرابعة"

تحديد عدد الصفوف بالجدول بصف واحد وبالتالى يقوم هذا الأمر بحذف صفوف البيانات السابقة إن وجدت

grid.Rows = 1

بداية حلقة تكرارية تستمر طالما لن نصل الى نهاية الملف المفتوح فى المنطقة ١

While Not EOF(1)

قراءة حقول السجل من الملف وتخزينها فى متغيرات ذاكرة

Input #1, mno, mname, myear, mkind, mvisual, mwindows, mexcel

فى حالة تساوى المتغيرين

If my = mname then

زيادة عدد صفوف الجدول بواحد لعرض بيانات السجل الحالى

grid.Rows = grid.Rows + 1

الانتقال داخل الصف الجديد وهو اخر صف فى الجدول

grid.Row = grid.Rows - 1

الانتقال داخل العمود التاسع

```
grid.Col = 8
grid.Text = mno
grid.Col = 7
grid.Text = mname

grid.Col = 6
grid.Text = myear
grid.Col = 5
grid.Text = mkind
grid.Col = 4
grid.Text = mvisual

grid.Col = 3
grid.Text = mwindows
grid.Col = 2
grid.Text = mexcel
```

كتابة الرقم

الانتقال داخل العمود الثامن

كتابة الاسم

الانتقال داخل العمود السابع

كتابة الفرقة

الانتقال داخل العمود السادس

كتابة النوع

الانتقال داخل العمود الخامس

كتابة درجة الفجول بيسك

الانتقال داخل العمود الرابع

كتابة درجة الويندوز

الانتقال داخل العمود الثالث

كتابة درجة الاكسل

الانتقال داخل العمود الثانى

```
grid.Col = 1
```

تخزين مجموع الدرجات فى متغير

```
msum = Val(mvisual) + Val(mwindows) + Val(mexcel)
```

كتابة مجموع الدرجات

```
grid.Text = msum
```

الانتقال داخل العمود الاول

```
grid.Col = 0
```

كتابة التقدير العام عن طريق استدعاء الدالة التي تستلم المجموع وتعود بالتقدير العام

```
grid.Text = g(msum)
end if
```

نهاية الحلقة التكرارية وهى تعنى نقل التحكم لبداية الحلقة التكرارية مرة اخرى

```
Wend
```

إغلاق الملف

```
Close #1
```

إظهار الجدول إذا كان عدد صفوفه اكبر من واحد أى عندما يحتوى الجدول على صفوف بيانات طلبة بالفعل

```
If grid.Rows > 1 Then grid.Visible = True
End Sub
```

فى حالة الضغط على اى مفتاح داخل مربع الرقم

```
Private Sub txtno_KeyPress(KeyAscii As Integer)
```

الانتقال إلى مربع الاسم عند الضغط علي مفتاح الإدخال

```
If KeyAscii = 13 Then txtname.SetFocus
End Sub
```

فى حالة الضغط على اى مفتاح داخل مربع الاسم

```
Private Sub txtname_KeyPress(KeyAscii As Integer)
```

الانتقال إلى مربع مادة الفجول بيسك عند الضغط على مفتاح الإدخال

```
If KeyAscii = 13 Then txtvisual.SetFocus
End Sub
```

فى حالة الضغط على اى مفتاح داخل مربع مادة الفجول بيسك

```
Private Sub txtvisual_KeyPress(KeyAscii As Integer)
```

الانتقال إلى مربع مادة الويندوز عند الضغط على مفتاح الإدخال

```
If KeyAscii = 13 Then txtwindows.SetFocus
End Sub
```

فى حالة الانتقال من مربع مادة الفجول بيسك

```
Private Sub txtvisual_LostFocus()
```

العودة إلى مربع مادة الفجول بيسك مرة أخرى فى حالة إدخال درجة اكبر من الدرجة القصوى (١٠٠) أو درجة اقل من الصفر

```
If Val(txtvisual) > 100 Or Val(txtvisual) < 0 Then
txtvisual.SetFocus

End Sub
```

فى حالة الضغط على اى مفتاح داخل مربع مادة الويندوز

```
Private Sub txtwindows_KeyPress(KeyAscii As Integer)
```

انتقال الى مربع مادة الاكسل عند الضغط على مفتاح الإدخال

```
If KeyAscii = 13 Then txtexcel.SetFocus
End Sub
```

فى حالة الانتقال من مربع مادة الويندوز

```
Private Sub txtwindows_LostFocus()
```

العودة إلى مربع مادة الويندوز مرة أخرى فى حالة إدخال درجة اكبر من الدرجة القصوى (٥٠) أو درجة اقل من الصفر

```
If Val(txtwindows) > 50 Or Val(txtwindows)< 0 Then
  txtwindows.SetFocus
 Endif
End Sub
```

فى حالة الضغط على اى مفتاح داخل مربع مادة الاكسل

```
Private Sub txtexcel_KeyPress(KeyAscii As Integer)
```

استدعاء إجراء الحفظ عند الضغط على مفتاح الإدخال

```
If KeyAscii = 13 Then cmdsave_Click
End Sub
```

فى حالة الانتقال من مربع مادة الاكسل

```
Private Sub txtexcel_LostFocus()
```

العودة إلى مربع مادة الاكسل مرة أخرى فى حالة إدخال درجة اكبر من الدرجة القصوى(٥٠) أو درجة اقل من الصفر

```
If Val(txtexcel) > 50 Or Val(txtexcel) < 0 Then
txtexcel.SetFocus
End Sub
```

فى حالة النقر على زر البحث عن الطالب الأول

```
Private Sub cmdfirst_Click()
```

الخروج من إجراء عرض المعلومات فى حالة عدم وجود الملف

```
If Dir$("data")  = "" Then MsgBox "الملف غير موجود" :Exit Sub
```

فتح الملف التتابعى للعرض

```
Open "data" For Input As #1
```

تحديد متغير رقمى لتخزين أكبر درجة

```
Max = 0
```

تحديد متغير حرفى لتخرين اسم صاحب أكبر درجة

```
mname  =""
```

تعبئة متغير الفرقة بالفرقة التى اختارها المستخدم للبحث عن أول طالب فيها

```
If optyear(0).Value = True Then my = "الأولى"
If optyear(1).Value = True Then my = "الثانية"
If optyear(2).Value = True Then my = "الثالثة"
If optyear(3).Value = True Then my = "الرابعة"
```

بداية حلقة تكرارية تستمر طالما لن نصل إلى نهاية الملف المفتوح فى المنطقة ١

```
While Not EOF(1)
```

قراءة حقول السجل من الملف وتخزينها فى متغيرات ذاكرة

```
Input #1, mno, mname, myear, mkind, mvisual, mwindows,
mexcel
```

تخزين مجموع الدرجات فى متغير

فى حالة تساوى الفرقة فى المتغيرين

```
If my = myear Then

msum = Val(mvisual) + Val(mwindows) + Val(mexcel)
```

تعبئة متغيرى اكبر درجة وصاحبها فى حالة اذا كان مجموع درجات الطالب الحالى أكبر من الدرجة الكبرى السابقة

```
If msum > Max Then Max = msum: mn = mname
End if
Wend
```

إغلاق الملف

```
Close #1
```

الخروج من الإجراء اذا لم يكن هناك درجة كبرى

```
If Max = 0 Then Exit Sub
```

ظهور رسالة بصاحب اكبر درجة تقديره

```
MsgBox mn & "    " & g(Max)
End Sub
```

فى حالة النقر على زر الخروج

Private Sub cmdend_Click()

إنهاء البرنامج

End
End Sub

دالة تحديد التقدير العام فهى تستلم مجموع الدرجات وتعطى التقدير العام

Function g(x)

تخزين نسبة النجاح فى متغير بعد ضربها فى ١٠٠ لتصبح النسبة رقم صحيح علما بأن المجموع الكلى لدرجات المواد الثلاث ٢٠٠ .

*p = x / 200 * 100*

فى حالة النسبة أكبر من ٩٠ تعبئة متغير التقدير بامتياز

If p > 90 Then g = "امتياز"

فى حالة النسبة من ٨٠ الى اقل من ٩٠ تعبئة متغير التقدير بجيد جدا

If p < 90 And p >= 80 Then g = "جيد جدا"

فى حالة النسبة من ٦٥ الى اقل من ٨٠ تعبئة متغير التقدير بجيد

If p < 80 And p >= 65 Then g = "جيد"

فى حالة النسبة من ٥٠ الى اقل من ٦٥ تعبئة متغير التقدير بمقبول

If p < 65 And p >= 50 Then g = "مقبول"

فى حالة النسبة اقل من ٥٠ تعبئة متغير التقدير بضعيف

If p < 50 Then g = "ضعيف'

نهاية الدالة

End Function

التدريب الخامس

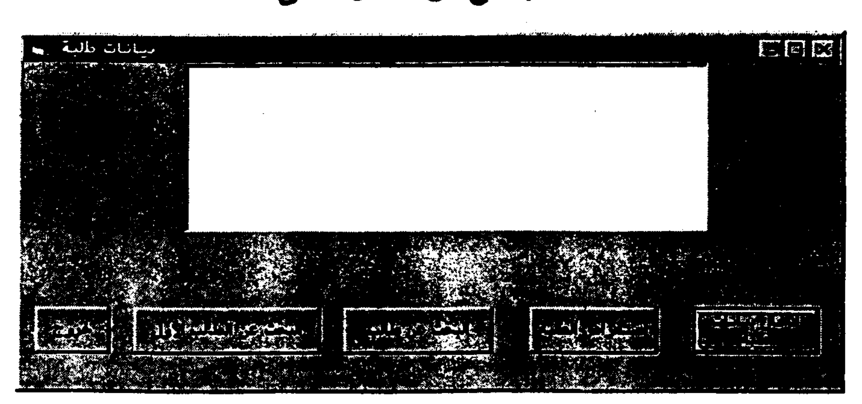
إدخال بيانات طلبة فرقة معينة بالكلية ودرجاتهم فى ثلاث مواد مع استعرضهم بياناتهم ومجموع هذه المواد والبحث عن طالب معين وعرض الطالب الأول على الفرقة

أولا : الأدوات المطلوبة

يتم إضافة الأدوات التالية على نموذج جديد :

- عدد ٥ زر أمر
- عدد ١ قائمة عرض

كما يتضح من الشكل التالى



ثانياً : الأكواد

تعريف متغير معرف عن طريق موديول ليمثل عدة الحقول التى تمثل سجل طالب ويتم إضافة الموديول من قائمة المشروع كما يلى

Project
Add Module

ثم كتابة ذلك داخل الموديول

```
Type stud
```

تعريف متغير رقمى صحيح لرقم الطالب

```
no As Integer
```

تعريف متغير حرفى حجمه ٢٠ حرف لاسم الطالب

```
na As String * 20
```

تعريف متغير حرفى حجمه ١٥ حرف لعنوان الطالب

```
ad As String * 15
```

تعريف متغير رقمى فردى يحتوى على كسور لدرجة المادة الاولى

```
d1 As Single
```

تعريف متغير رقمى فردى يحتوى على كسور لدرجة المادة الثانية

```
d2 As Single
```

تعريف متغير رقمى فردى يحتوى على كسور لدرجة المادة الثالثة

```
d3 As Single
```

نهاية تعريف النوع

```
End Type
```

يتم فتح مكان الكود الخاص بالنافذة

الإعلان عن متغير استنادا إلى النوع السابق تعريفه أعلاه وذلك حتى يكون متاحا لكل اجراءات النافذة ويكتب هذا السطر فى منطقة الإعلان الموجودة فى قمة نافذة الكود

```
Dim s As stud
```

فى حالة تحميل النافذة

```
Private Sub Form_Load()
```

عنوان النافذة

```
 Caption = "بيانات طلبة"
```

فتح ملف عشوائى

```
 Open "student" For Random As #1 Len = Len(s)
End Sub
```

عند النقر على زر إدخال بيانات الطلبة

```
Private Sub cmdinput_Click()
تعبئة متغير بعدد الطلبة داخل المراد إدخالهم
 n = Val(InputBox("أدخل عدد الطلاب المراد إدخالهم"))
بداية حلقة تكرارية بعدد الطلاب المراد إدخالهم
 For i = 1 To n
إدخال البيانات المطلوبة داخل حقول المتغير المعرف
  s.no = InputBox("ادخل رقم الطالب")
  s.na = InputBox("ادخل الاسم")
  s.ad = InputBox("ادخل العنوان")
  s.d1 = Val(InputBox("ادخل درجة المادة الاولى"))
  s.d2 = Val(InputBox("ادخل درجة المادة الثانية"))
  s.d3 = Val(InputBox("ادخل درجة المادة الثالثة"))
تخزين البيانات داخل الملف العشوائى
  Put #1, i, s
نهاية الحلقة التكرارية وتعنى إعادة التحكم لبداية الحلقة
 Next
End Sub
```

عند النقر على زر استعراض الطلبة

```
Private Sub cmdshow_Click()
تنظيف القائمة من البيانات السابقة إن وجدت
 lst.Clear
بداية حلقة تكرارية بعدد السجلات الموجودة بالملف
عدد السجلات =عدد حروف الملف /عدد حروف السجل
 For i = 1 To LOF(1) / Len(s)
تعبئة المتغير بحقول السجل الحالى بالملف
 Get #1, i, s
```

إضافة بيانات الحقول داخل القائمة

```
lstAddItem Str(s.no) & "  " & s.na & "  " & "  " & s.ad & "
" & Str(s.d1) & "  " & Str(s.d2) & "  " & Str(s.d3) & "  " &
Str(s.d1 + s.d2 + s.d3)
```

نهاية الحلقة التكرارية وتعنى إعادة التحكم لبداية الحلقة

```
Next
End Sub
```

فى حالة النقر على زر البحث عن طالب

```
Private Sub cmdseekname_Click()
```

تعبئة الاسم المراد البحث عنه داخل متغير

```
mname = InputBox("ادخل اسم الطالب المراد البحث عنه") _
```

الخروج من الإجراء إذا كان متغير الاسم خالى

```
If mname = "" Then Exit Sub
```

تنظيف القائمة من البيانات السابقة إن وجدت

```
lst.Clear
```

بداية حلقة تكرارية بعدد السجلات الموجودة بالملف

عدد السجلات =عدد حروف الملف /عدد حروف السجل

```
For i = 1 To LOF(1) / Len(s)
```

تعبئة المتغير بحقول السجل الخاص بالطالب التالى

```
Get #1, i, s
```

إضافة بيانات الحقول داخل القائمة فى حالة إذا حقل الاسم (بعد حذف المسافات الموجودة قبله وبعده) يساوى متغير الاسم

```
If Trim(s.na) = mname Then lstAddItem Str(s.no) & "  " & s.na
& "  " & "  " & s.ad & "  " & Str(s.d1) & "  " & Str(s.d2) & "
" & Str(s.d3) & "  " & Str(s.d1 + s.d2 + s.d3(
Next
End Sub
```

فى حالة النقر على زر البحث عن الطالب الأول

```
Private Sub cmdfirst_Click()
```

تحديد متغير رقمى لتخزين أكبر درجة

```
Max = 0
```

تحديد متغير حرفى لتخزين اسم صاحب أكبر درجة

```
mname = ""
```

بداية حلقة تكرارية بعدد السجلات الموجودة بالملف
عدد السجلات =عدد حروف الملف /عدد حروف السجل

```
For i = 1 To LOF(1) / Len(s)
```

تعبئة المتغير بحقول السجل الخاص بالطالب التالى

```
 Get #1, i, s
```

تعبئة مجموع درجات الثلاث مواد داخل متغير

```
 msum = s.d1 + s.d2 + s.d3
```

تعبئة متغيرى اكبر درجة وصاحبها فى حالة إذا كان مجموع درجات الطالب الحالى أكبر من الدرجة الكبرى السابقة

```
 If msum > Max Then Max = msum: mname = s.na
Next
```

الخروج من الإجراء إذا لم يكن هناك درجة كبرى

```
If Max = 0 Then Exit Sub
```

ظهور رسالة بصاحب اكبر درجة ودرجته

```
 MsgBox mname & "   " & Max
End Sub
Private Sub cmdend_Click()
```

الخروج من البرنامج

```
 End
End Sub
```

التدريب السادس

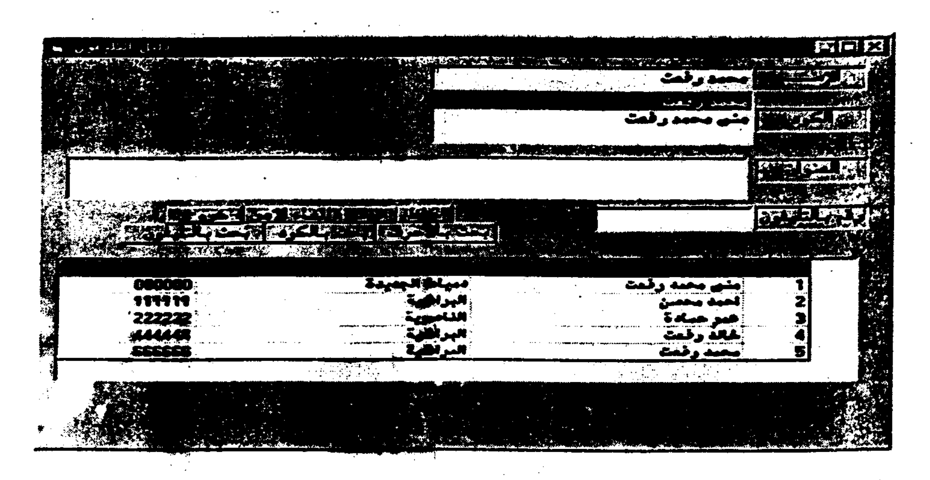
تصميم برنامج دليل تليفون باستخدام ملفات قاعدة البيانات

أولا: الأدوات المطلوبة

- ٧ زر أمر — ١ قائمة
- ٤ مربع نصى — ٤ مربع عنوان
- ١ جدول (*grid*) — ١ اداة قاعدة بيانات (مخفية) *Data1*

يتم تغير خاصية (*Datasource*) من نافذة خصائص الجدول بحيث تظهر بها *Data1*

كما يتضح من الشكل التالى



ثانياً : تصميم قاعدة البيانات

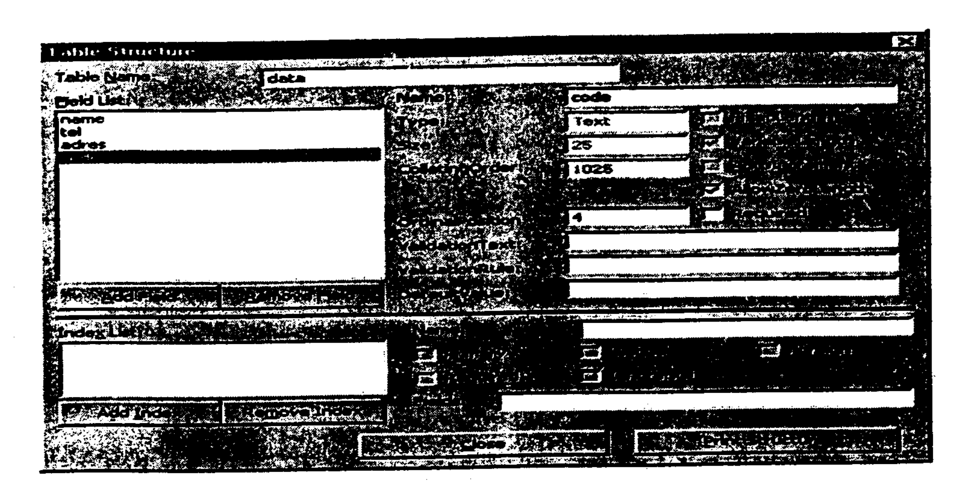
عن طريق فتح قائمة

Add – Ins

Visual data manager

ثم فتح ملف جديد باسم (*Cust*)بداخله جدول باسم (*Data*) بحيث يظهر هذا الشكل

هيكل جدول البيانات (*Data*)



توصيف حقول الجدول

Field	Type	Size
Name	Text	35
Tel	Text	15
Adres	Text	100
Code	Text	15

ثالثاً : الأكواد

في حالة تحميل النافذة

```
Private Sub Form_Load()
```

تحديد أبعاد النافذة من القمة

```
Top = 300
```

الارتفاع

```
Height = 6000
```

من اليسار

```
Left = 150
```

العرض

```
Width = 9300
```

تسمية النافذة

```
Caption = "دليل التليفون"
```

عدم تمكين مفتاح الحذف لأنه لا توجد بيانات حاليا

```
cmddel.Enabled = False
```

الانتقال إلى فهرس البرنامج

```
ChDir App.Path
```

إخفاء جدول عرض البيانات لأنه لا توجد بيانات حاليا به

```
grid.Visible = False
```

تحديد ارتفاع قائمة لأسماء .

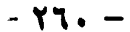
```
lst.Height = 1000
End Sub
```

في حالة الضغط على مفتاح داخل مربع الكود

```
Private Sub txtcode_KeyDown(KeyCode As Integer, Shift As Integer)
```

الانتقال إلى مربع العنوان عند الضغط مفتاح الإدخال

```
If KeyCode = 13 Then txtadres.SetFocus
End Sub
```




عند تنشيط مربع الاسم

```
Private Sub txtname_GotFocus()
```

استدعاء إجراء تنظيف المربعات النصية من البيانات

```
 cmdclear
```

لتحديد الاسم المكتوب داخل مربع الاسم

```
 txtname.SelStart = 0
 txtname.SelLength = Len(txtname.Text)
End Sub
```

فى حالة النقر فى قائمة الأسماء

```
Private Sub lst_Click()
```

تعبئة مربع الاسم بالاسم المنقور(المضغوط) عليه في قائمة الأسماء

```
 txtname.Text = lst.List(lst.ListIndex)
End Sub
```

فى حالة الضغط على أى مفتاح داخل قائمة الأسماء

```
Private Sub lst_KeyDown(KeyCode As Integer, Shift As Integer)
```

فى حالة الضغط على مفتاح الإدخال داخل قائمة الأسماء ويكون مربع الاسم ممتلئ

```
 If KeyCode = 13 And txtname.Text <> "" Then
```

إخفاء قائمة الأسماء

```
  lst.Visible = False
```

استدعاء إجراء تعبئة المربعات النصية بالبيانات

```
  fill
 End If
End Sub
```

فى حالة الضغط على أى مفتاح داخل مربع الاسم

```
Private Sub txtname_KeyDown(KeyCode As Integer, Shift As
Integer)
```

حالة الضغط على مفتاح الإدخال وكان مربع الاسم به بيانات

```
 If KeyCode = 13 And txtname.Text <> "" Then
```

فتح ملف البيانات داخل متغير

Set dbq = Workspaces(0).OpenDatabase(pdir & "cust.mdb")

إنشاء منطقة استفسار داخل متغير

Set myq = dbq.CreateQueryDef("")

كتابة الاستفسار (عرض كل الأسماء من داخل جدول البيانات مرتبة حسب الأسماء)

myq.SQL = "SELECT name FROM data ORDER BY [name]"

فتح جدول بناتج الاستفسار داخل متغير

Set filt = myq.OpenRecordset(dbOpenSnapshot)

البحث فى حقل الاسم عن الاسم المدخل فى المربع النصى من قبل المستخدم

filt.FindFirst "name = '" & txtname.Text & "'"

إذا وجد الاسم المطلوب يتم إغلاق مناطق البحث وتحرير أماكنها فى الذاكرة واستدعاء إجراء تعبئة المربعات النصية والخروج من الإجراء الحالى

```
If Not filt.NoMatch Then
  filt.Close: myq.Close: dbq.Close: Set filt = Nothing
  Set myq = Nothing: Set dbq = Nothing: fill
  Exit Sub
End if
```

وإذا لم يجد الاسم المطلوب يستعرض الأسماء التى تشترك معا فى الحروف المدخلة فى مربع الاسم كما يلى :

- إيجاد عدد الحروف الموجودة فى مربع الاسم بعد حذف المسافات فى البداية والنهاية

nc = Len(Trim(txtname.Text))

– البحث فى حقل الاسم عن الأسماء التى تشترك فى الحروف المدخلة فى المربع النصى

```
filt.FindFirst "left(name," & nc & ")= '" & Trim(txtname.Text) & "'"
```

– إذا لم يجد أى اسم ينطبق عليه الشرط السابق معنى ذالك ان هناك اسم جديد نريد إدخاله فننتقل إلى مربع إدخال الكود وكذلك يتم إغلاق مناطق البحث وتحرير أماكنها فى الذاكرة والخروج من الإجراء الحالى

```
If filt.NoMatch Then
  filt.Close: myq.Close: dbq.Close: Set filt = Nothing
  Set myq = Nothing: Set dbq = Nothing: txtcode.SetFocus
  Txtcode.setfocus:Exit Sub
End if
```

إلى هنا يكون لدينا أسماء يمكن عرضها فى القائمة لاختيار احدها
تنظيف القائمة من أى بيانات سابقة

```
lst.Clear
```

بداية حلقة تكرارية لتعبئة القائمة بما وجد من أسماء وتستمر الحلقة طالما لم نصل إلى نهاية الملف

```
Do While Not filt.EOF()
```

تعبئة القائمة بالاسم الذى ينطبق عليه الشرط

```
 If Left(filt!Name, nc) = Trim(txtname.Text) Then
   lst.AddItem filt!Name
 End if
```

الانتقال للسجل التالى

```
 filt.MoveNext
Loop
```

إغلاق مناطق البحث وتحرير أماكنها فى الذاكرة

```
filt.Close: myq.Close: dbq.Close: Set filt = Nothing
Set myq = Nothing: Set dbq = Nothing
```

إذا كانت القائمة بها أسماء (أى عدد عناصرها لا يساوى صفر)

```
If lst.ListCount <> 0 Then
```

اظهر قائمة البيانات

```
 lst.Visible = True
```

نشط القائمة

```
 lst.SetFocus
```

نشط العنصر الأول فى القائمة

```
 lst.ListIndex = 0
```

املأ مربع الاسم بأول اسم فى القائمة

```
 txtname.Text = lst.List(0)
```

إذا كانت القائمة بها اسم واحد فقط اخفى القائمة واستدعى برنامج تعبئة المربعات النصية ببيانات الاسم المذكور

```
 If lst.ListCount = 1 Then lst.Visible = False: fill
 End If
End If
End Sub
```

فى حالة الضغط على أى مفتاح داخل مربع العنوان

```
Private Sub txtadres_KeyDown(KeyCode As Integer,Shift As Integer)
```

فى حالة الضغط على مفتاح الإدخال يتم الانتقال إلى مربع التليفون

```
 If KeyCode = 13 Then txttel.SetFocus
End Sub
```

فى حالة تنشيط المربع النصى الخاص بالتليفون

```
Private Sub txttel_GotFocus()
```

لتحديد النص المكتوب داخل مربع التليفون

```
 txttel.SelStart = 0
 txttel.SelLength = Len(txttel.Text)
End Sub
```

في حالة الضغط على أى مفتاح داخل المربع النصى الخاص بالتليفون

```
Private Sub txttel_KeyDown(KeyCode As Integer, Shift As
Integer)
```

نشط زر الحفظ فى حالة الضغط على مفتاح الإدخال

```
 If KeyCode = 13 Then cmdsave.SetFocus
End Sub
```

فى حالة النقر على مربع الحفظ

```
Private Sub cmdsave_Click()
```

إذا كان الاسم غير مكتوب فى المربع النصى الخاص بالاسم يتم اظهار رسالة (أدخل الاسم) ثم الخروج من الإجراء

```
If txtname.Text = "" Then MsgBox "أدخل الاسم":
txtname.SetFocus: Exit Sub
```

فتح ملف البيانات داخل متغير

```
Set Db = Workspaces(0).OpenDatabase("cust.mdb")
```

فتح جدول البيانات داخل متغير

```
Set rs = Db.OpenRecordset("data", dbOpenDynaset)
```

كتابة شرط البحث

```
mycr = "name = '" & txtname.Text & "'"
```

البحث داخل جدول البيانات باستخدام الشرط

```
rs.FindFirst mycr
```

فى حالة وجوده

```
If Not rs.NoMatch Then
```

افتح السجل للتعديل

```
 rs.Edit
```

فى حالة عدم وجوده

```
Else
```

افتح سجل جديد

```
 rs.AddNew
End If
```

كتابة البيانات من المربع النصى إلى جدول البيانات

```
rs!Name = txtname.Text
rs!adres = txtadres.Text
rs!code = txtcode.Text
rs!tel = txttel.Text
rs.Update
```

استدعاء إجراء تنظيف المربعات النصية من البيانات

```
cmdclear
```

الانتقال داخل مربع الاسم

```
txtname.SetFocus
```

استعراض البيانات بعد الحفظ

```
cmdseek_Click 0
```

إظهار رسالة "تم الحفظ" فى الملف

```
MsgBox "تم الحفظ"
End Sub
```

فى حالة النقر على زر إلغاء الأمر

```
Private Sub cmdcancel_Click()
```

استدعاء إجراء تنظيف المربعات النصية من البيانات

```
cmdclear
```

الانتقال داخل مربع الاسم

```
txtname.SetFocus
End Sub
```

إجراء عرض البيانات داخل المربعات النصية

```
Sub fill()
```

فتح ملف البيانات فى متغير

```
Set dbq = Workspaces(0).OpenDatabase("cust.mdb")
```

إنشاء منطقة للاستفسار فى متغير

```
Set myq = dbq.CreateQueryDef("")
```

شروط الاستفسار (اختر كل السجلات من جدول البيانات بشرط أن حقل الاسم يحتوى على الاسم الموجود بمربع الاسم

```
myq.SQL = "SELECT * FROM cust.data WHERE" &
            "name = '" & txtname.Text & "'"
```

افتح الملف بالشروط السابقة فى متغير

```
Set filt = myq.OpenRecordset(dbOpenSnapshot)
```

فى حالة وجود الاسم فى الملفات (أى عندما لا يصل البحث الى نهاية الملف)

```
If Not filt.EOF Then
```

اكتب البيانات داخل المربعات النصية

```
 txtcode.Text = filt!code
 txtname.Text = filt!Name
 txtadres.Text = filt!adres
 txttel.Text = filt!tel
```

تمكين مفتاح الحذف لوجود بيانات يمكن حذفها

```
 cmddel.Enabled = True
 Else
```

استدعاء إجراء تنظيف المربعات النصية من البيانات

```
 cmdclear
 End If
End Sub
```

فى حالة النقر على مربع الحذف

```
Private Sub cmddel_Click()
```

فتح ملف البيانات داخل متغير

```
Set Db = Workspaces(0).OpenDatabase("cust.mdb")
```

فتح الجدول داخل متغير

```
Set rs = Db.OpenRecordset("data", dbOpenDynaset)
```

البحث عن الاسم داخل الحقل الموجود بالجدول

```
rs.FindFirst "name = '" & txtname.Text & "'"
```

فى حالة وجوده

If Not rs.NoMatch Then

احذفه من الجدول

rs.Delete

استدعاء إجراء تنظيف المربعات النصية من البيانات

cmdclear

الانتقال داخل مربع الاسم

txtname.SetFocus

استعراض البيانات بعد الحذف

cmdseek_Click 0
End If
End Sub

فى حالة النقر على أحد مربعات البحث الثلاث
مع العلم أن مربعات البحث الثلاث لها اسم واحد ولكنها مختلفة فى خاصية الفهرس

Private Sub cmdseek_Click(index As Integer)

فتح ملف البيانات داخل متغير

Set dbq = Workspaces(0).OpenDatabase(pdir & "cust.mdb")

انشاء منطقة استفسار داخل متغير

Set myq = dbq.CreateQueryDef("")

التفرع سيتم بناء على قيمة الخاصية (index)
index هنا تمثل خاصية من خواص أداة أزرار الأوامر الثلاثة نظرا لأنهم الثلاث يحملون نفس الاسم ولذا فهذه الخاصية هى التى تفرق بينهم
بداية التفرع

Select Case index

إذا كانت قيمة الخاصية = ٠ أى أن شرط البحث سيكون على الاسم

Case 0

إذا كان مربع الاسم غير خالى قم بتعبئة متغير الشرط بالشرط

If txtname.Text <> "" Then

```
    cond = " where left(name," & Len(Trim(txtname)) & ") = '"
&
    Left(txtname, Len(Trim(txtname))) & "'"
  End if
```

إذا كانت قيمة الخاصية = ١ أى أن شرط البحث سيكون على الكود

```
 Case 1
```

إذا كان مربع الكود غير خالى قم بتعبئة متغير الشرط بالشرط

```
  If txtcode.Text <> "" Then
    cond = " where left(code," & Len(Trim(txtcode)) & ") = '"
&
    Left(txtcode, Len(Trim(txtcode))) & "'"
  End if
```

إذا كانت قيمة الخاصية = ٢ أى أن شرط البحث سيكون على التليفون

```
 Case 2
```

إذا كان مربع التليفون غير خالى قم بتعبئة متغير الشرط بالشرط

```
if txttel.Text <> "" Then
    cond = " where left(tel," & Len(Trim(txttel)) & ") = '" &
    Trim(txttel) & "'"
End if
```

نهاية التفرع

```
End Select
```

اختيار كل السجلات من جدول البيانات طبقا للشرط أن وجد (cond) مرتبة حسب الاسم

```
myq.SQL = "SELECT tel as التليفون,adres as العنوان,
              name as الاسم,code as الكود FROM " & "cust.data "
              & cond & " order by name"
```

فتح جدول بناتج الاستفسار السابق فى متغير

```
Set filt = myq.OpenRecordset(dbOpenSnapshot)
```

الكود	الشرح
`If Not filt.EOF Then`	إذا احتوى جدول الاستفسار على بيانات
`grid.Visible = True`	إظهار جدول عرض البيانات
`Else`	وإلا
`grid.Visible = False`	إخفاء جدول عرض البيانات
`Exit Sub`	الخروج من الإجراء
`End If`	
`Set Data1.Recordset = filt`	ربط أداة البيانات بجدول الاستفسار
`grid.Row = 0`	الانتقال للصف الأول
`grid.Col = 3`	الانتقال للعمود الرابع
`grid.ColAlignment(3) = 6`	محاذاة بيانات العمود المذكور ناحية اليمين
`grid.ColWidth(3) = 700`	تحديد عرض العمود المذكور
`grid.Col = 2`	وهكذا لباقى الأعمدة
`grid.ColAlignment(2) = 6`	
`grid.ColWidth(2) = 3000`	
`grid.Col = 1`	
`grid.ColAlignment(1) = 6`	
`grid.ColWidth(1) = 3000`	

```
grid.Col = 0
grid.ColAlignment(0) = 6
grid.ColWidth(0) = 1500
```

استدعاء إجراء تنظيف المربعات النصية من البيانات

```
cmdclear
End Sub

Sub cmdclear()
```

تنظيف المربعات النصية من البيانات

```
txtname.Text = " "
txtadres.Text = " "
txttel.Text = " "
txtcode.Text = " "
```

عدم تمكين مفتاح الحذف لعدم وجود بيانات معروضة

```
cmddel.Enabled = False
End Sub
```

فى حالة النقر على زر الخروج

```
Private Sub cmdend_Click()
```

إنهاء البرنامج

```
End
End Sub
```

التدريب السابع

برنامج قائمة رئيسية لمعالجة بيانات الطلاب بكلية التربية النوعية بدمياط

يحتوي هذا البرنامج على عدة برامج فرعية و هي :

١- برنامج لإضافة بيانات طلاب الكلية في ملف تتابعي
٢- برنامج لعرض بيانات الطلاب بطرق مختلفة
٣- برنامج لعمل صيانة للملف أو نسخة منه في مكان آخر
٤- برنامج لإلغاء بيانات طالب ما
٥- برنامج لطباعة بيانات الطلاب على الطابعة
٦- برنامج لإنهاء التعامل مع القائمة الرئيسية

و فيما يلي محتويات كل برنامج و شرحه على حدة :

أولاً :

محتويات نافذة ال form لعمل القائمة الرئيسية :

١- تحتوي على labels لكتابة رأس البرنامج
٢- تحتوى على ٦ commands و ذلك لتنفيذ البرامج الفرعية من خلالها

و ذلك كما في الشكل التالي :

نظام لمعالجة بيانات الطلاب
بكلية التربية النوعية بدمياط

القائمة الرئيسية

(1)	برنامج إضافة	(4)	برنامج إلغاء بيانات
(2)	برنامج عرض	(5)	برنامج طباعة
(3)	برنامج صيانة	(6)	إنهاء البرنامج

ثانياً :

في برنامج إضافة بيانات طلاب كلية التربية النوعية بدمياط في ملف تتابعي و تشتمل هذه البيانات على الآتي (رقم الطالب – اسم الطالب - درجة مقدمة الحاسب – درجة قواعد بيانات– درجة البرامج الجاهزة – درجة الإحصاء – درجة أمن البيانات)

في نافذة الكود الخاصة بالزر الأول يكتب الآتي :

```
Open "student.txt" for append as #1
V_code = inputbox(" ادخل رقم الطالب ")

A:
P = inputbox (" هل تريد إضافة بيانات سجل جديد ")
If p="y" or p="Y" then
Print #1 , v_code
N = inputbox ("ادخل الاسم ")
Print #1 , n
D1 = inputbox (" ادخل درجة مقدمة الحاسب")
Print #1 , d1
```

```
D2 = inputbox (" ادخل درجة قواعد البيانات ")
Print #1 , d2
D3 = inputbox (" ادخل درجة البرامج الجاهزة ")
Print #1 , d3
D4 = inputbox (" ادخل درجة الإحصاء ")
Print #1 , d4
D5 = inputbox (" ادخل درجة أمن البيانات ")
Print #1 , d5
V_code = v_code +1
Goto a
Else
Goto b
Endif

B:
Close #1
```

شرح أوامر البرنامج :

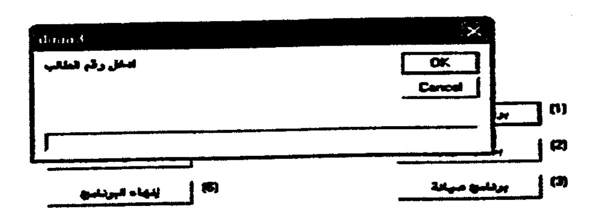
الأمر الأول : هو أمر فتح الملف و الإضافة فيه
الأمر الثاني : هو أمر خاص بإدخال رقم أول طالب نريد إدخال بياناته
الأمر الثالث : و هو خاص بتنفيذ ما يليه
الأمر الرابع : رسالة لمعرفة هل تريد إدخال بيانات أم لا
الأمر الخامس : هو شرط لتنفيذ الإضافة أم لا
الأمر السادس : يستخدم للكتابة في الملف المفتوح

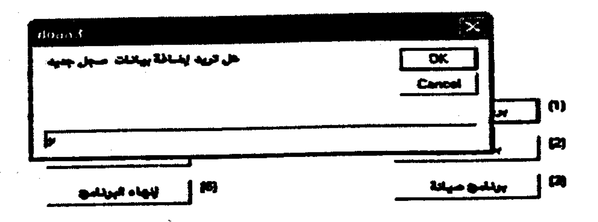
و هكذا للباقي إلي الأمر الثامن عشر يتم ظهور مجموعة من الرسائل ليتم إضافة البيانات المختلفة داخل الملف المفتوح

الأمر التاسع عشر : و ذلك لإضافة طالب جديد يتم الإدخال لبياناته و غير ذلك يتم إغلاق الملف

تنفيذ البرنامج :

عند الضغط على زر الإضافة يتم ظهور هذه الشاشة و يتم إدخال رقم الطالب ثم الضغط على ok و هكذا تتوالى ظهور النوافذ للإضافة.





ثالثاً :

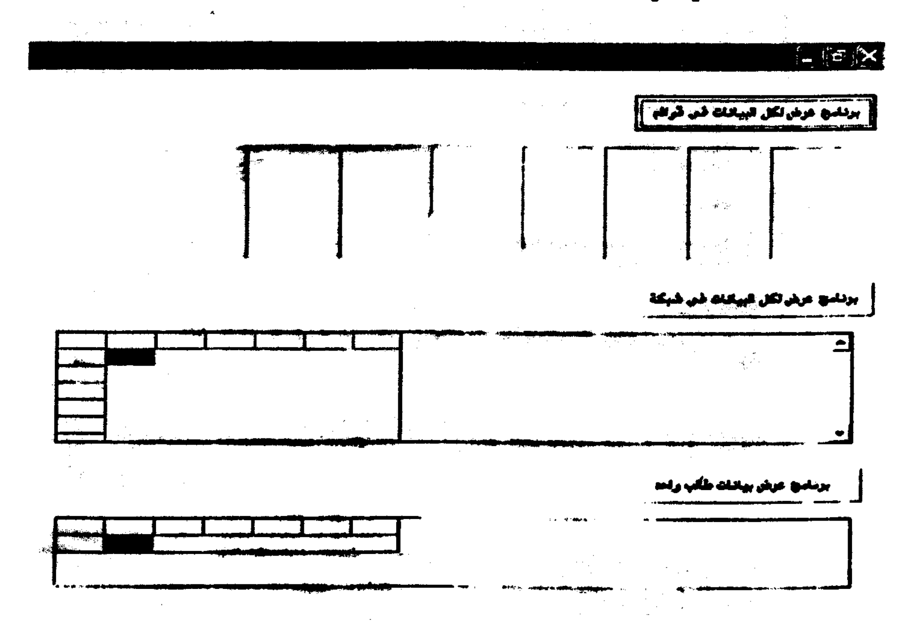
في برامج عرض بيانات طلاب كلية التربية النوعية بدمياط من الملف التتابعي و تشتمل هذه البيانات على عرض الآتي (رقم الطالب – اسم الطالب – درجة

مقدمة الحاسب – درجة قواعد بيانات– درجة البرامج الجاهزة – درجة الإحصاء – درجة أمن البيانات)

عند الضغط على هذا الزر فانه ينتقل إلى نافذة form أخرى وذلك لتنفيذ إحدى الخيارات التالية :

١- عرض بيانات كل الطلاب في قوائم
٢- عرض بيانات كل الطلاب في شبكة
٣- عرض بيانات طالب ما

و يظهر ذلك كما يلي في نافذة ال form2 :



و لكي يتم الانتقال من form1 إلى form2 يتم كتابة الأوامر التالية في نافذة الكود الخاصة بالزر الثاني :

```
Load form2
Form1.hide
Form2.show
```

و ذلك كما يلي :

```
Command2                                Click

Private Sub Command2_Click()
Load Form2
Form1.Hide
Form2.Show
End Sub

Private Sub Command3_Click()
Open " student " For Input As #1
FileCopy "d:\program files\microsoft visual studio\vb98\do
FileCopy "d:\program files\microsoft visual studio\vb98\do
End Sub

Private Sub Command4_Click()
Load Form3
Form1.Hide
Form3.Show

End Sub

Private Sub Command5_Click()
Printer.EndDoc

End Sub

Private Sub Command7_Click()
```

و في زر عرض بيانات كل الطلاب في قوائم الموجود في نافذة form2 يتم كتابة الأوامر التالية :

```
Open "student.txt" For Input As #1
List1.Clear: List2.Clear
List3.Clear: List4.Clear
List5.Clear: List6.Clear
```

```
List7.Clear

a1 = "رقم الطالب"
List1.AddItem a1
List1.AddItem"__ - _____"

a2 = "اسم الطالب"
List2.AddItem a2
List2.AddItem"__ _ _____"

a3 ="درجة مقدمة الحاسب"
List3.AddItem a3
List3.AddItem"_______________"

a4 = "درجة قواعد البيانات"
List4.AddItem a4
List4.AddItem"________________"

a5 = "درجة البرامج الجاهزة"
List5.AddItem a5
List5.AddItem"_________________"

a6 = "درجة الإحصاء "
List6.AddItem a6
List6.AddItem"__________"

a7 = "درجة أمن البيانات"
List7.AddItem a7
```

```
List7.AddItem"___________________"

While Not EOF(1)
Input #1, v_code
List1.AddItem v_code

Input #1, n
List2.AddItem n

Input #1, d1
List3.AddItem d1

Input #1, d2
List4.AddItem d2

Input #1, d3
List5.AddItem d3

Input #1, d4
List6.AddItem d4

Input #1, d5
List7.AddItem d5

Wend
Close #1
```

و ذلك كما يلي :

```
Command1                    Click
Private Sub Command1_Click()
Open " student " For Input As #1
List1.Clear: List2.Clear
List3.Clear: List4.Clear
List5.Clear: List6.Clear
List7.Clear

a1 = " الرقم"
List1.AddItem a1
List1 AddItem "_____"

a2 = " الاسم"
List2.AddItem a2
List2.AddItem "_____"

a3 = " المقدمة "
List3.AddItem a3
List3.AddItem "_____"

a4 = " القواعد "
List4.AddItem a4
List4.AddItem "_____"

a5 = " البرامج "
List5.AddItem a5
List5.AddItem "_____"
```

شرح أوامر البرنامج :

الأمر الأول : فتح الملف و ذلك لعرض البيانات الموجودة به .
الأمر الثاني إلى الأمر الخامس : و ذلك لجعل القوائم التي يتم عرض البيانات بها نظيفة (أي خالية من البيانات) .
الأمر السادس إلى الأمر السادس و العشرون : في هذه الأوامر طريقة لوضع العناوين للقوائم التي يتم عرض البيانات بها .
الأمر السابع و العشرون إلى آخر البرنامج : و ذلك لتنفيذ الجمل الخاصة بعرض البيانات و التي تكون محصورة بين جملتي wend و while و مشروطة بعدم الوصول إلى نهاية الملف ثم بعدها يتم غلق الملف .

تنفيذ البرنامج :

عند الضغط على الزر الخاص بعرض بيانات كل الطلاب من الملف في قوائم
يتم ظهور البيانات في قوائم كالتالي :

و في زر عرض بيانات كل الطلاب في شبكة الموجود في نافذة form2
يتم كتابة الأوامر التالية :

```
Open "student.txt" for input as #1

Grid1.row=0
Grid1.col=0
```

```
A1="رقم الطالب"
Grid1.text=a1

Grid1.col=1
A2="اسم الطالب"
Grid1.text=a2

Grid1.col=2
A3="درجة مقدمة الحاسب"
Grid1.text=a3

Grid1.col=3
A4="درجة قواعد البيانات"
Grid1.text=a4

Grid1.col=4
A5="درجة البرامج الجاهزة"
Grid1.text=a5
Grid1.col=5
A6="درجة الإحصاء"
Grid1.text=a6

Grid1.col=6
A7="درجة أمن البيانات"
Grid1.text=a7

Grid1.row=1

While not eof(1)
```

```
For I = 0 to 6
Grid1.col=I
Input #1 , x
Grid1.text=x
Next
Grid1.row = grid1.row+1
Wend

Close #1
```

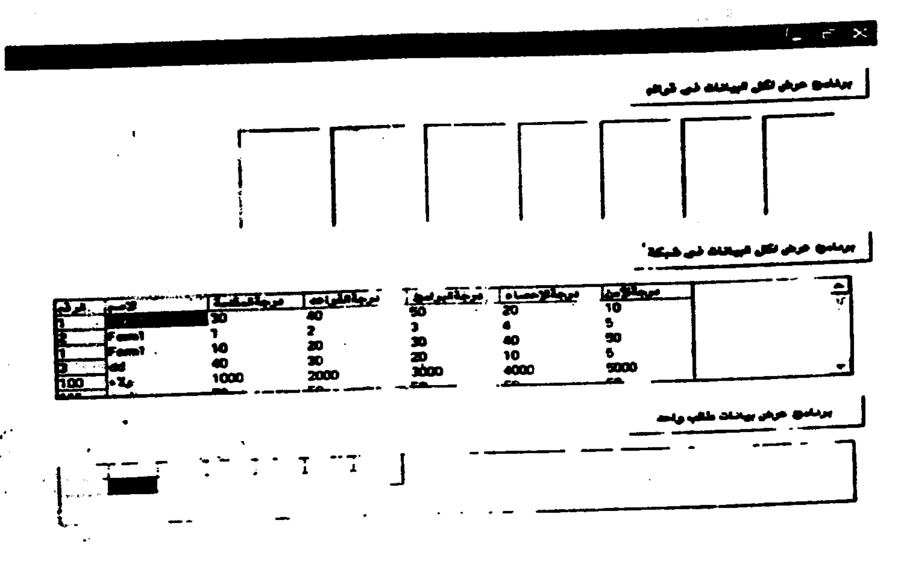
شرح أوامر البرنامج :

الأمر الأول : فتح الملف
الأمر الثاني : تحديد الصف الأول في الشبكة
الأمر الثالث : تحديد العمود الأول في الشبكة
من الأمر الرابع إلى الأمر الثالث و العشرون : جمل لتحديد العناوين في الصف الأول و الأعمدة من صفر إلى السادس في grid1 .
الأمر الرابع و العشرون : لزيادة الصف بمقدار ١

باقي البرنامج أوامر لعرض بيانات الملف حتى يصل إلى نهايته ثم إغلاق الملف .

تنفيذ البرنامج :

عند الضغط على الزر الخاص بعرض كل البيانات في شبكة من الملف يتم ظهور البيانات في شبكة كالتالي :



و في زر عرض بيانات طالب واحد في شبكة الموجود في نافذة form2 يتم كتابة الأوامر التالية :

```
Open "student.txt" For Input As #1

10:

10 code = InputBox("أدخل رقم السجل")

Grid2.Row = 0
Grid2.Col = 0
a1 = "الرقم"
Grid2.Text = a1
Grid2.Col = 1
a2 = "الإسم "
Grid2.Text = a2
```

```
Grid2.Col = 2
a3 = " درجة المقدمة"
Grid2.Text = a3
Grid2.Col = 3
a4 = " درجة القواعد"
Grid2.Text = a4
Grid2.Col = 4
a5 = " درجة البرامج "
Grid2.Text = a5
Grid2.Col = 5
a6 = " درجة الإحصاء"
Grid2.Text = a6
Grid2.Col = 6
a7 = " درجة الأمن"
Grid2.Text = a7
Grid2.Row = 1

While Not EOF(1)
For I = 0 To 6
Grid2.Col = I
Input #1, v1_code
Grid2.Text = v1_code
Next
Wend

p1 = InputBox("هل تريد عرض بيانات سجل آخر؟")
If p1 = "y" Or p1 = "Y" Then

GoTo 10
```

```
End If

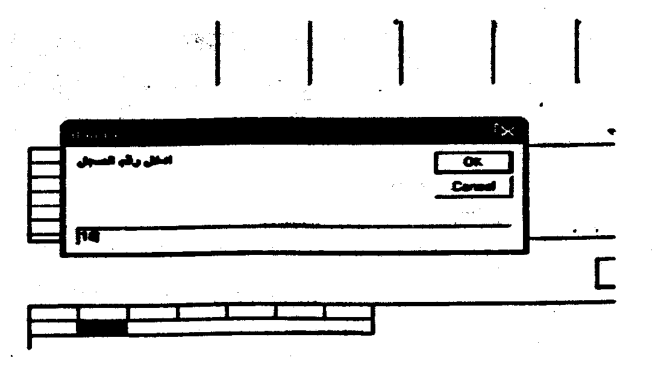
Close #1
```

شرح أوامر البرنامج :
الأمر الأول : فتح الملف باستخدام أمر open .
الأمر الثاني : لتحديد رقم السجل المراد عرض بياناته
الأمر الثالث : تحديد الصف الأول في الشبكة
الأمر الرابع : تحديد العمود الأول في الشبكة
من الأمر الخامس إلى الأمر الرابع و العشرون : جمل لتحديد العناوين في الصف الأول و الأعمدة من صفر إلى السادس في grid1 .
الأمر الخامس و العشرون : لزيادة الصف بمقدار ١

باقي البرنامج أوامر لعرض بيانات الملف حتى يصل إلى نهايته ثم إغلاق الملف مع العلم أنه توجد رسالة تفيد بمعرفة هل تريد عرض بيانات سجل آخر .

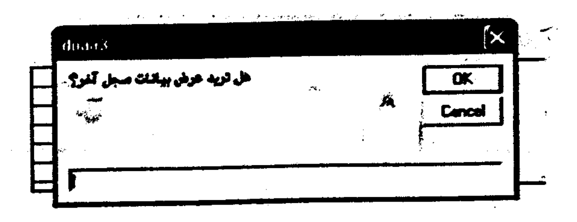
تنفيذ البرنامج :

عند الضغط على الزر الخاص بعرض بيانات طالب واحد في شبكة من الملف يتم ظهور البيانات في شبكة كالتالي :



 MR





رابعاً :

في برنامج صيانة بيانات طلاب كلية التربية النوعية بدمياط أي عمل نسخة من الملف التتابعي على أي وسط تخزيني آخر
يتم كتابة الأوامر التالية في نافذة code :

```
FileCopy "d:\program files\microsoft visual
studio\vb98\student.txt", "c:\studentbk.txt "
```

شرح أوامر البرنامج :

الأمر الأول : filecopy و هو خاص بعمل نسخة من الملفات في مكان آخر وهنا معناه عمل نسخة من الملف المسمى student و امتداده txt الموجود على وحدة الإدارة الثابتة الثانية و على الدليل الفرعي المستوى الأول و الثاني و الثالث إلى وحدة الإدارة الثابتة الأولى و على الدليل الرئيسي لها.

تنفيذ البرنامج :

يتم عمل نسخة من الملف إلى وحدة الإدارة الثابتة الأولى وعلى الدليل الرئيسي لها و بذلك يكون لدينا نسخة أخرى من الملف.

خامساً :

في برنامج إلغاء بيانات طالب ما من الملف التتابعي
يتم كتابة الأوامر التالية في نافذة code ·

```
Open "student.txt" For Input As #1

10:
v1_code = InputBox("أدخل رقم السجل")

Grid1.Row = 0
Grid1.Col = 0
a1 = "رقم الطالب"
Grid1.Text = a1
Grid1.Col = 1
a2 = "اسم الطالب"
Grid1.Text = a2
Grid1.Col = 2
a3 = "درجة مقدمة الحاسب"
Grid1.Text = a3
Grid1.Col = 3
a4 = "درجة قواعد البيانات"
Grid1.Text = a4
Grid1.Col = 4
a5 = "درجة البرامج الجاهزة"
Grid1.Text = a5
Grid1.Col = 5
a6 = "درجة الإحصاء"
Grid1.Text = a6
```

```
Grid1.Col = 6
a7 = "درجة أمن البيانات"
Grid1.Text = a7
Grid1.Row = 1

While Not EOF(1)
For I = 0 To 6
Grid1.Col = I
Input #1, v1_code
Grid1.Text = v1_code
Next
Wend

p1 = InputBox("هل تريد حذف بيانات السجل؟")
If p1 = "y" Or p1 = "Y" Then

For I = 0 To 6
Grid1.Col = I
Grid1.Text = ""
Next

End If

p1 = InputBox("هل تريد عرض بيانات سجل اخر ثم حذفه ؟")
If p1 = "y" Or p1 = "Y" Then
GoTo 10

End If
```

Close #1

شرح أوامر البرنامج :

الأمر الأول : فتح الملف باستخدام أمر open

الأمر الثاني : لتحديد رقم السجل المراد عرض بياناته ثم حذفها

الأمر الثالث : تحديد الصف الأول في الشبكة

الأمر الرابع : تحديد العمود الأول في الشبكة

من الأمر الخامس إلى الأمر الرابع و العشرون : جمل لتحديد العناوين في الصف الأول و الأعمدة من صفر إلى السادس في grid1.

الأمر الخامس و العشرون : لزيادة الصف بمقدار ١

من الأمر السادس و العشرون إلى الأمر الثاني و الثلاثون : خاص بعرض بيانات السجل الذي تم تحديده لمشاهدتها قبل الإلغاء.

من الأمر الثالث و الثلاثون إلى الأمر التاسع و الثلاثون : و هو خاص بظهور رسالة تأكيدية للإلغاء و في حالة الموافقة يتم الإلغاء للسجل المحدد من قبل

أما باقي البرنامج فهو ظهور رسالة تسأل عن سجل آخر يتم عرضه ثم إلغاؤه ثم إغلاق البرنامج .

تنفيذ البرنامج :

يتم ظهور نافذة لتحديد رقم السجل المراد إلغاؤه ثم عرض هذا السجل و ظهور رسالة تأكيدية أخرى لعملية الإلغاء

و السؤال بعد ذلك عن أي سجل آخر و ذلك كما يلي:

برنامج إلغاء بيانات طالب

رقم	اسم	درجة	درجة	درجة	درجة	درجة
30		20	30	50	40	10

daaa3

هل تريد حذف بيانات المسجل ؟

OK

Cancel

برنامج إلغاء بيانات طالب

رقم	اسم	درجة	درجة	درجة	درجة	درجة

daaa3

هل تريد عرض بيانات سجل آخر ثم حذفه ؟

OK

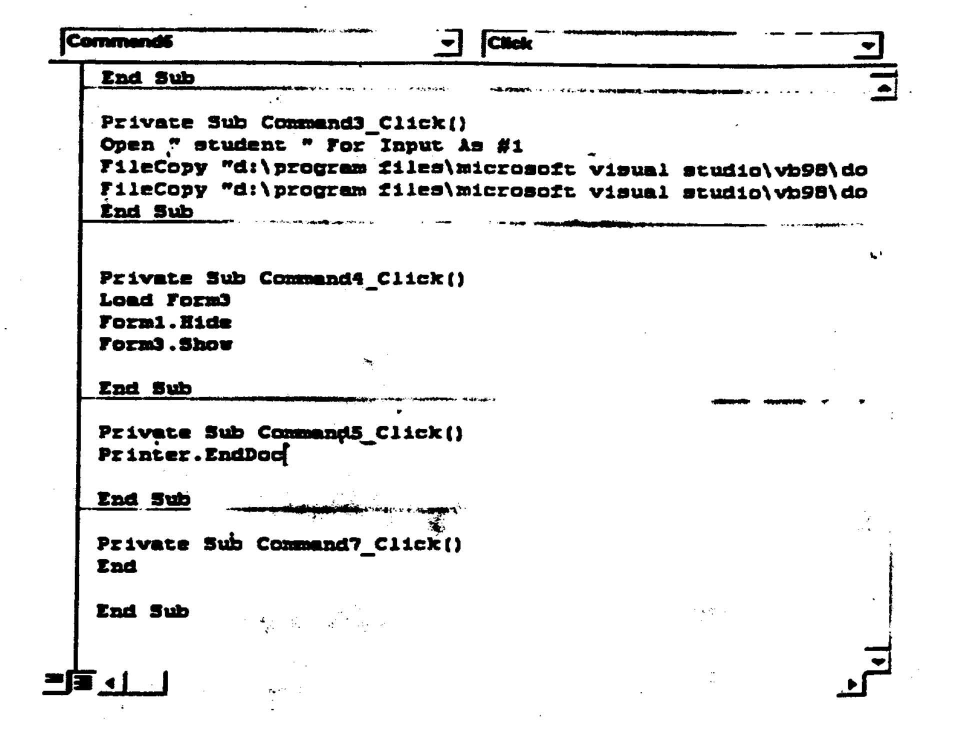
Cancel

سادساً :

في برنامج طباعة البيانات من الملف التتابعي
يتم كتابة الأمر التالي في نافذة code :

```
Printer.EndDoc
```

و ذلك كما يلي :



```
Command6                                  Click
End Sub

Private Sub Command3_Click()
Open " student " For Input As #1
FileCopy "d:\program files\microsoft visual studio\vb98\do
FileCopy "d:\program files\microsoft visual studio\vb98\do
End Sub

Private Sub Command4_Click()
Load Form3
Form1.Hide
Form3.Show

End Sub

Private Sub Command5_Click()
Printer.EndDoc

End Sub

Private Sub Command7_Click()
End

End Sub
```

سابعاً :

في برنامج إنهاء التعامل مع الملف التتابعي
يتم كتابة الأمر التالي في نافذة code :

End

و ذلك كما يلي :

```
Command7                                   Click
End Sub

Private Sub Command3_Click()
Open " student " For Input As #1
FileCopy "d:\program files\microsoft visual studio\vb98\do
FileCopy "d:\program files\microsoft visual studio\vb98\do
End Sub

Private Sub Command4_Click()
Load Form3
Form1.Hide
Form3.Show

End Sub

Private Sub Command5_Click()
Printer.EndDoc

End Sub

Private Sub Command7_Click()
End

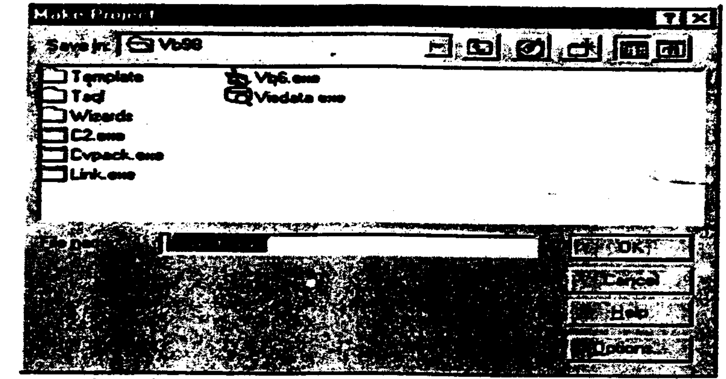
End Sub
```

خطوات عمل حزم البرامج

١- عمل نسخة تنفيذية من أى برنامج عن طريق قائمة ملف من نافذة الفيجول بيسك كما يلى :

File

Make Project1.exe



ويمكن تغير اسم الملف التنفيذى قبل الضغط على زر *Ok* كما فى هذا الشكل

ولابد أن تحتوى حزمة البرنامج على الملفات الآتية قبل توزيعها

- مكتبات لغة الفجول بيسك (*Dll , Ocx , Vbx ....*)
- مكتبات قاعدة البيانات المستخدمة (*Ado* أو *Dao ....*)
- الملفات التنفيذية (*Exe*)
- ملفات البيانات (*Dbf* أو *Mdb ....*)

٢- تكوين حزم البرامج داخل ملفات نوعها (*Cab*) بالإضافة إلى ملفات تحميل البرنامج (*Setup.exe , Setup.lst*) عن طريق البرنامج *Package & Deployment Wizard*

Programs

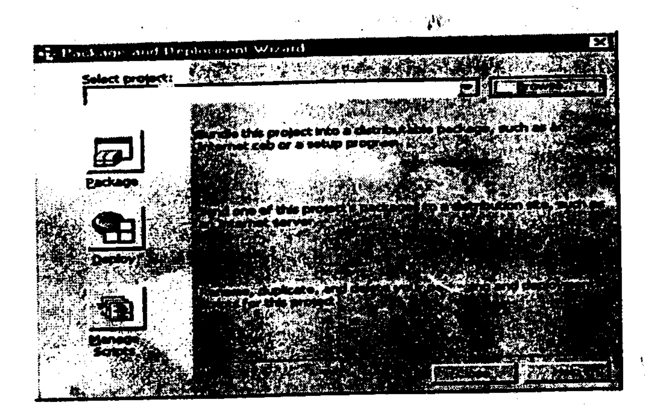
Microsoft Visual Studio 6.0

Microsoft Visual Studio 6.0 Tools

Package & Deployment Wizard

Start

كما يظهر بهذا الشكل نكتب اسم المشروع ثم ننقر زر *Package*



ثم نستمر فى الانتقال من نافذة إلى أخرى حتى يكتمل تكوين حزمة البرنامج.

المراجع العربية:

١ – برمجة قواعد البيانات Visual Basic 6 ، عزيز اسبر، علي سليمان، دار الكتب العلمية للنشر والتوزيع، عابدين القاهرة.

٢ – موسوعة مبرمجى فيجوال بيسك ٦، محمد أسعد نشاوى، أحمد وضاح عطار، دار الكتب العلمية للنشر والتوزيع، عابدين القاهرة.

المراجع الأجنبية:

1- Fundamentals Of Programming In basic,.1981, Robert G. Nickerson, Winthrop Publishers ,inc, Cambridge.
2- Programming with basic –byrons, Gottfried,1984,England.
3- PC Operations Guide, Basic to The 16 Bit PC ,1985,USA,
4- Gw-Basic Interpreters User's Guide 1986 By Commodore Electronics Limited Germany.

البرامج الجاهزة :

5- Microsoft Visual Basic Ver. 4 , 6 .

مواقع الانترنت

www.alain.cc/vb/showthread.php?mode=hybrid&t=44 -52k
www.arabteam2000-forum.com
www.egmix.com/board/showthread.php?t=3640 -44k
vb4arab.com/vb/showthread.php?t=8087 - 41k
www.geocities.com/f_bal3awi/23.htm - 74k

فهرست

الفصل الأول – مدخل إلى البيسيك المرئي Visual Basic

الفصل الثاني – التعامل مع قوائم برنامج Visual Basic 6.0

الفصل الثالث – التعامل مع نافذة Toolsbox & Form

الفصل الرابع – نافذة الخصائص Properties

الفصل الخامس – تابع النوافذ الأخرى

الفصل السادس – الإجراءات والتكرار والتفرع

الفصل السابع - التدريبات العملية

** (تم بحمد الله) **